(19)日本国特許庁（ＪＰ）　(12) **公開特許公報（Ａ）**　(11)特許出願公開番号
特開2002－281065
（P2002－281065A）
(43)公開日　平成14年９月27日(2002.9.27)

| (51)Int.Cl.[7] | 識別記号 | ＦＩ | テーマコード*(参考) |
|---|---|---|---|
| Ｈ０４Ｌ　12/56 | | Ｈ０４Ｌ　12/56 | Ｚ　５Ｋ０３０ |

審査請求　未請求　請求項の数32　書面　外国語出願　（全 59 頁）

(21)出願番号　特願2001－197685(P2001－197685)
(22)出願日　平成13年６月29日(2001.6.29)
(31)優先権主張番号　０９／６０７８０７
(32)優先日　平成12年６月30日(2000.6.30)
(33)優先権主張国　米国（ＵＳ）

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(72)発明者　山本　彰
アメリカ国カリフォルニア州クパチーノ
ルッペルプレイス1202
(72)発明者　山神　憲司
アメリカ国カリフォルニア州ロスガトス
カイルニベル108
(74)代理人　100075096
弁理士　作田　康夫

Ｆターム(参考)　5K030 GA11 HA08 HC01 HC14 JT06
KA02

(54)【発明の名称】　データの完全性を伴いデータネットワークに接続される記憶装置システム

(57)【要約】
【解決手段】通信ネットワークによってリモート記憶装置コンポーネントに結合されたローカル記憶装置コンポーネントを備える記憶装置システム。通信ネットワークは、データパケットが送信されたものと同じ順序でのデータパケットの受信を保証できないことにより特徴づけられる。
【効果】本発明に従った方法は、当該通信ネットワークの性質にもかかわらず、受信された要求を処理する適切な順序づけを保証する。

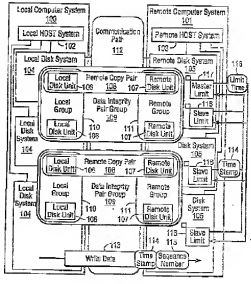



FIG. 1

【特許請求の範囲】

【請求項１】　記憶装置システムであって、

第１の記憶装置コンポーネントを有する第１のコンピュータシステムと、

第２の記憶装置コンポーネントを有する第２のコンピュータシステムとを含み、

第１および第２の記憶装置コンポーネントはデータネットワークによってデータを交換するように構成されており、

第１のコンピュータシステムは、データのブロックを第１の記憶装置コンポーネントに書込み、データパケットを第２のコンピュータシステムに伝送するためにプログラムコードで構成されたメモリを有しており、データパケットはデータブロック、タイムスタンプおよびシーケンス番号を含むものであり、

第２のコンピュータシステムは、第１のコンピュータシステムからデータパケットを受信し、データパケットに含まれているタイムスタンプおよびシーケンス番号に基づき候補データパケットを選択し、候補データパケットを第２の記憶装置システムに書込むようにプログラムコードにより構成されたメモリを有しており、

ここにおいて、第１の記憶装置コンポーネントに書込まれるデータのブロックは、第１の記憶装置コンポーネントにおけると同じ順序で第２の記憶装置コンポーネントに書込まれるものである、システム。

【請求項２】　第２のメモリがさらに、タイムスタンプの中からそれらの対応するシーケンス番号に基づきリミットタイムスタンプを取得し、それらの対応するタイムスタンプをリミットタイムスタンプと比較することによってデータパケットの中から候補データパケットを選択するようにプログラムコードにより構成されている、請求項１記載のシステム。

【請求項３】　データネットワークがコネクションレス型ネットワークである、請求項１記載のシステム。

【請求項４】　データパケットがそれらが送信されたものと同じ順序で受信されると保証できないものとしてデータネットワークが特徴づけられる、請求項１記載のシステム。

【請求項５】　データネットワークがワイドエリアネットワークである、請求項４記載のシステム。

【請求項６】　第１の記憶装置コンポーネントが複数の第１のデータ記憶装置よりなり、第２の記憶装置コンポーネントが複数の第２のデータ記憶装置よりなり、第１のデータ記憶装置の各々は第２のデータ記憶装置の１個と対応しており、ここにおいて、第１のデータ記憶装置の１個に格納されたデータが対応する第２のデータ記憶装置の１個にも格納される、請求項１記載のシステム。

【請求項７】　第１の記憶装置コンポーネントが複数の第１のディスクシステムよりなり、第２の記憶装置コンポーネントが複数の第２のディスクシステムよりなり、第１のディスクシステムは各々第２のディスクシステムの１個以上と関係づけられており、ここにおいて、第１のディスクシステムのうちの１個に格納されたデータが関係する１個以上の第２のディスクシステムにも格納される、請求項１記載のシステム。

【請求項８】　第１のディスクシステムの各々が複数の第１のディスク装置よりなり、第２のディスクシステムの各々が複数の第２のディスク装置よりなり、第１のディスク装置の各々は第２のディスク装置の１個と関係づけられている、請求項７記載のシステム。

【請求項９】　第１のディスク装置は各々、第１のディスク装置が属する第１のディスクシステムとは独立に、第２のディスク装置のうちの１個と関係づけられている、請求項８記載のシステム。

【請求項１０】　ローカルシステムに含まれるデータをリモートシステムにバックアップする方法であって、

データのブロックをローカルデータ記憶装置に書込むことと、

データパケットをリモートシステムに送信することと、

データパケットはデータのブロック、タイムスタンプおよびシーケンス番号を含むものであり、

ローカルシステムからデータパケットを受信することと、

そのデータのブロックがリモートデータ記憶装置に書込まれるデータパケットを、データパケットのシーケンス番号およびタイムスタンプに基づき選択することとを含む、方法。

【請求項１１】　次のデータパケットのシーケンス番号を増分することをさらに含む、請求項１０記載の方法。

【請求項１２】　データパケットを選択することが、タイムスタンプの中からそれらの関係するシーケンス番号に基づきリミットタイムスタンプを取得することと、それらの関係するタイムスタンプをリミットタイムスタンプと比較することによってデータパケットの中からそのデータパケットを選択することを含む、請求項１０記載の方法。

【請求項１３】　ローカルデータ記憶装置が複数のローカルディスク装置よりなり、リモートデータ記憶装置が複数のリモートディスク装置よりなり、各ローカルディスク装置がリモートコピーペアを規定するためにリモートディスク装置の１個と対にされている、請求項１０記載の方法。

【請求項１４】　ローカルディスク装置にデータの複数のブロックを書込むことと、各リモートディスク装置がその関係する複数のデータパケットに基づくシーケンス番号のリストを有するようにリモートディスク装置に複数のデータパケットを送信することとをさらに含み、該方法はさらに、シーケンス番号の各リストについて、シーケンス番号の最も長い連続を取得することと、その最も長い連続から最高値のシーケンス番号を取得すること

と、最高値のシーケンス番号に対応するタイムスタンプを取得することと、それによって、タイムスタンプのリストを生成することとを含み、該方法はさらに、タイムスタンプのリストの中で最早タイムスタンプに基づきデータパケットを選択することを含む、請求項１３記載の方法。

【請求項１５】　ローカルデータ記憶装置が複数のローカルディスクシステムよりなり、リモートデータ記憶装置が複数のリモートディスクシステムよりなり、各ローカルディスクシステムが１個以上のリモートディスクシステムと関係づけられており、ここにおいて、ローカルディスクシステムの１個以上に格納されたデータが、関係する１個以上のリモートディスクシステムにも格納される、請求項１０記載の方法。

【請求項１６】　ローカルディスクシステムの各々が複数のローカルディスク装置よりなり、リモートディスクシステムの各々が複数のリモートディスク装置よりなり、ローカルディスク装置の各々がリモートディスク装置の１個と関係づけられている、請求項１５記載の方法。

【請求項１７】　各ローカルディスク装置が、ローカルディスク装置が属するローカルディスクシステムとは独立に、リモートディスク装置の１個と関係づけられている、請求項１６記載の方法。

【請求項１８】　データのブロックをローカルデータ記憶装置に書込むこととデータパケットをリモートシステムに送信することが非同期に実行される、請求項１０記載の方法。

【請求項１９】　データパケットがコネクションレス型データネットワークによって送信される、請求項１０記載の方法。

【請求項２０】　データパケットが、それらが送信されたものと同じ順序で行先に着信すると保証できないものとして特徴づけられるデータネットワークによってデータパケットが送信される、請求項１０記載の方法。

【請求項２１】　データネットワークがワイドエリアネットワークである、請求項２０記載の方法。

【請求項２２】　複数のローカルデータ記憶装置よりなるローカル記憶装置システムにおいて、ローカル記憶装置システムのデータと、複数のリモートデータ記憶装置よりなるリモート記憶装置システムにバックアップする方法であって、該方法が、

各ローカルデータ記憶装置において、それに書込まれるデータブロックを受信することと、

各ローカルデータ記憶装置において、データブロック、タイムスタンプおよびシーケンス番号よりなるデータパケットをリモートデータ記憶装置の１個に伝送することと、

リモートデータ記憶装置において、ローカルデータ記憶装置から複数のデータパケットを受信することと、ここにおいて、各リモートデータ記憶装置は、その関係する複数のデータパケット、シーケンス番号のリストおよび、関係するデータパケットからのタイムスタンプのリストを有するものであり、

各リモートデータ記憶装置において、シーケンス番号の中で最も長い連続を識別し、最も長い連続の最高値のシーケンス番号のデータパケットを取得することと、

各リモートデータ記憶装置において、得られたデータパケットから最早タイムスタンプを取得することと、

得られたタイムスタンプのうちで最早のものをリミットタイムとして選択することと、

各リモートデータ記憶装置において、最早タイムスタンプを有する候補データパケットを選択することと、

タイムスタンプがリミットタイムより早い候補データパケットの中からデータパケットを選択することとを含む、方法。

【請求項２３】　ローカル記憶装置システムが１個以上のローカルディスクシステムよりなり、各ローカルディスクシステムが１個以上のローカルディスク駆動装置よりなり、リモート記憶装置システムが１個以上のリモートディスクシステムよりなり、各リモートディスクシステムが１個以上のリモートディスク駆動装置よりなり、各ローカルディスク駆動装置がリモートコピーペアを規定するためにリモートディスク駆動装置の１個と関係づけられている、請求項２２記載の方法。

【請求項２４】　各ローカルデータ記憶装置がローカルディスク駆動装置の１個であり、各リモートデータ記憶装置がリモートディスク駆動装置のうちの１個であり、ここにおいて、各リモートコピーペアと関係づけられたシーケンス番号が存在する、請求項２３記載の方法。

【請求項２５】　受信された複数のデータパケットがリモートコピーペアに従ってグループ化される、請求項２４記載の方法。

【請求項２６】　シーケンス番号が、共通のリモートコピーペアを有するローカルおよびリモートディスクシステムの各対と関係づけられている、請求項２３記載の方法。

【請求項２７】　複数のデータパケットの各々が、それが送信された元のローカルディスクシステムに基づきグループ化される、請求項２６記載の方法。

【請求項２８】　各リモートコピーペアが複数のデータの完全性ペアの１個と関係づけられており、ここにおいて、シーケンス番号が、共通のデータの完全性ペアグループを有するローカルおよびリモートディスクシステムの各対と関係づけられている、請求項２３記載の方法。

【請求項２９】　データアクセス方法であって、

データ転送要求を付与することと、データ転送要求はシーケンス番号を含むものであり、

データ転送要求をローカルシステムからリモートシステムに伝送することと、

リモートシステムにおいて、データのブロックを含んでいるエントリの待ち行列を付与することと、データ転送要求は待ち行列のターゲットエントリに向けられており、
リモートシステムにおいて、データ転送要求のシーケンス番号をシーケンス番号カウンタの現在値と比較することと、
シーケンス番号がシーケンス番号カウンタと等しくない場合、シーケンス番号とシーケンス番号カウンタとの差に基づくエントリの数だけ待ち行列をトラバースすることによって待ち行列のターゲットエントリへのアクセスを得ることと、
シーケンス番号がシーケンス番号カウンタに等しい場合、ターゲットエントリへのアクセスを得るために待ち行列の一方の端にアクセスすることと、
ターゲットエントリでデータ転送要求を実行することとを含む、方法。

【請求項３０】　リモートシステムが磁気テープ記憶装置システムを含み、データ転送要求が磁気テープ記憶装置システムへの読出しおよび書込み要求である、請求項２９記載の方法。

【請求項３１】　データ転送要求が読出し要求であり、待ち行列のデータのブロックが読出し要求を満たすために使用される、請求項２９記載の方法。

【請求項３２】　データ転送要求が、新しいエントリとして待ち行列に挿入される書込みデータを含む書込み要求であり、新しいエントリがターゲットエントリの前または後に挿入される、請求項２９記載の方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、データ記憶装置システム、詳細には、データネットワーク環境においてデータ記憶装置システムのデータの完全性を維持することに関する。

【０００２】

【従来の技術】従来、データ処理システムは、高速な高信頼性データバスによって各自の関係するデータ記憶装置システムにアクセスできる。しかし、その機会は、ネットワーク通信の広範囲に及ぶ用途が拡大し続けるにつれ、ますます利用可能になっている。ＩＰ（インターネットプロトコル）は、ＴＣＰ／ＩＰ（伝送制御プロトコル／インターネットプロトコル）ネットワークが確立される基本的なパケットデリバリサービスを提供する。ＩＰは、十分に規定されたプロトコルであり、それゆえ、ＩＰを用いてサーバシステムが記憶装置システムとデータを交換し、記憶装置システムが他の記憶装置システムとデータを交換する、ネットワークベースのデータ記憶アクセスのためのトランスポート／ネットワーク層を提供する自然な候補である。

【０００３】しかし、ＩＰの性質は、データ記憶アクセスシステムの領域におけるいくつかの独自な問題を呈する。第一に、ＩＰはコネクションレス型プロトコルである。これは、ＩＰがデータの伝送に先んじてエンドツーエンド接続を確立するために制御情報を交換しないことを意味する。ＩＰは誤り検出・回復機構をまったく含まない。従って、ＩＰは接続されたネットワークにデータを送達するためには信頼できるが、そのデータが正しく受信されたか、またはそれが送信された順序でデータが受信されているかを保証するためのいかなる機構も存在しない。コネクション型サービスが要求される場合、ＩＰは、接続を確立するためにより高次の階層プロトコルに頼る。

【０００４】デュアルリモートコピー機能が必要とされるデータ記憶装置システムにおいて、ＩＰベースの伝送は問題を呈する。リモートコピー機能は、一次データ記憶装置での災害回復を実現するという目的で、リモートサイトにおける一次データ記憶装置のリアルタイムコピーを可能にする。この機能がその目的を果たすために、データの完全性を保証することが重要である。同期および非同期の２種類のリモートコピーがある。

【０００５】同期型リモートコピーでは、ローカルＨＯＳＴによるその関係するローカルディスクシステムへの書込み要求は、書込まれたデータがローカルディスクシステムからリモートディスクシステムに転送された後まで完了しない。従って、同期型コピーの場合、ローカルおよびリモートディスクシステム間のデータの完全性を保証することは容易である。

【０００６】非同期型リモートコピーでは、ローカル日ＯＳＴによる書込み要求は、ローカルディスクがリモートディスクへのその転送を完了する前に、完了する。名称が暗示する通り、ローカルディスクからリモートディスクへの転送動作が完了しているかどうかにかかわらず、制御はローカルＨＯＳＴに返される。従って、非同期型コピー動作の間のデータの完全性は、リモートディスクのデータがローカルディスクにおけると同じ順序で書込まれるように、リモートディスクシステムでのデータの正しい着信順序に依存する。

【０００７】これを実現するために、ローカルディスクシステムは、リモートディスクに送られるデータとともにタイムスタンプを含める。リモートにおいてデータはそのタイムスタンプに従って書込まれる。従って例えば、リモートディスクがタイムスタンプ７：００を備えるデータを受信した時には、それはすでに、タイムスタンプが７：００より前である全部のデータを受信している。

【０００８】

【発明が解決しようとする課題】しかし、ＩＰベースのネットワークにおいて、パケットが乱順で着信できる場合、７：００のタイムスタンプを有するデータパケットは、それ以前のタイムスタンプを有するデータパケット

が先行することもあればしないこともあり得る。その結果、伝送プロトコルがＩＰのようなコネクションレス型トランスポートモデルに基づく場合、リモートディスクシステムにおいてデータの完全性を保証することは難しい。

【０００９】ＩＰが磁気テープシステムで使用された時には別の問題が生じる。磁気テープへの読出しおよび書込み動作はシーケンシャルであり、従って、アドレス指定は固定である。従ってデータパケットが乱順でリモートサイトに着信した場合、データは間違った順序でテープに書込まれる。破壊された記憶装置システムをテープから復元するための以降の回復動作は結果として破損データを生じるであろう。

【００１０】信頼できるＩＰベースのデータ回復システムを提供するための必要性が存在する。

【００１１】

【課題を解決するための手段】本発明に従ったデータ記憶装置システムは、ローカル記憶装置コンポーネントおよびリモート記憶装置コンポーネントよりなる。ローカル記憶装置コンポーネントに書込まれるデータは、データネットワークによってリモート記憶装置コンポーネントにデータパケットで送られる。

【００１２】データパケットは、ローカル記憶装置コンポーネントで書込まれるデータのコピー、タイムスタンプおよびシーケンス番号を含む。複数のそうしたデータパケットがリモート記憶装置コンポーネントで受信される。データパケットは、各データパケットに関係するシーケンス番号およびタイムスタンプに基づきリモート記憶装置コンポーネントでの書込みのために選択される。

【００１３】本発明の１実施形態において、ローカルおよびリモート記憶装置コンポーネントはそれぞれ、複数のローカルおよびリモートディスク装置として構成される。各ローカルディスク装置はリモートディスク装置と関係づけられる。そのような各対はリモートコピーペアと呼ばれる。この実施形態では、各リモートコピー単位は関係するシーケンス番号を有する。

【００１４】本発明の別の実施形態において、ローカルディスク装置はローカルディスクシステムにグループ化される。同様に、各リモートディスク装置はリモートディスクシステムにグループ化される。本発明のこの実施形態では、少なくとも１個の共通のリモートコピーペアを有する各対のローカルおよびリモートディスクシステムについてシーケンス番号が存在する。

【００１５】本発明の別の実施形態において、リモートコピーペアはデータの完全性ペアグループにグループ化される。この実施形態では、少なくとも１個のデータの完全性ペアグループを共通に有するローカルおよびリモートディスクシステムの各対についてシーケンス番号が存在する。

【００１６】本発明の教示は、添付する詳細な説明を以下の図面とともに検討することによって容易に理解することができる。

【００１７】

【発明の実施の形態】図１を参照すれば、本発明の１実施形態に従ったコンピュータシステムが示されている。ローカルコンピュータシステム１００は、少なくともローカルＨＯＳＴシステム１０２および少なくともローカルディスクシステム１０４を含む。リモートコンピュータシステム１０１は少なくともリモートディスクシステム１０５を含む。リモートＨＯＳＴシステム１０３は必ずしもリモートコンピュータシステム１０１に必要なわけではない。個々のローカルディスクシステム１０４は通信路１１２によってリモートディスクシステム１０５と接続される。通信路１１２は、伝送されたデータパケットが必ずしもそれらが送られた順序で着信するわけではないことを特徴とするネットワークである。ＩＰベースのネットワークはこの振舞いを示す。一般に、コネクションレス型ネットワークはこの振舞いを示す。例えば、ワイドエリアネットワーク（ＷＡＮ）はＩＰに基づくことができる。しかし、本発明はＷＡＮに限定されるものではない。

【００１８】ローカルディスクシステム１０４は、リモートディスクシステム１０５において維持される関係するリアルタイムコピーを有する少なくともローカルディスク装置１０６を含む。リモートディスクシステム１０５は、ローカルディスク装置１０６のリアルタイムコピーを含んでいる少なくともリモートディスク装置１０７を含む。ローカルディスク装置１０６およびリモートディスク装置１０７よりなるペアは、リモートコピーペア１０８と称する。その中においてデータの完全性が保証されなければならないリモートコピーペア１０８のグループは、データの完全性ペアグループ１０９と称する。データの完全性ペアグループ１０９に属するローカルディスク装置１０６のグループは、データの完全性ローカルディスクグループ１１０と称する。同様に、リモートシステムでは、１個のデータの完全性ペアグループ１０９に属するリモートディスク装置１０７のグループは、データの完全性リモートディスクグループ１１１と称する。

【００１９】データの完全性ローカルディスクグループ１１０は、単一のローカルディスクシステム１０４から、または、２個以上のローカルディスクシステム１０４からのローカルディスク装置１０６を含み得る。その結果、ローカルディスクシステム１０４の構成要素であるローカルディスク装置１０６は、１個以上のデータの完全性ペアグループ１０９に見ることができる。データの完全性リモートディスク装置グループ１１１は、１個以上のリモートディスクシステム１０５に属するリモートディスク装置１０７より構成され得る。従って、リモートディスクシステム１０５は、異なるデータの完全性

ペアグループ１０９に属するリモートディスク装置１０７を有することができる。

【００２０】ローカルＨＯＳＴシステム１０２の動作の経過の間に、データはローカルディスクシステム１０４に書込まれる必要があるであろう。ローカルＨＯＳＴは、ローカルディスクシステム１０４に記憶される「書込みデータ」１１３を転送する。ローカルディスクシステム１０４はまた、書込みデータ１１３をデータ回復のためにリモートディスクシステム１０５にも送る。本発明によれば、ローカルディスクシステム１０４がリモートディスクシステム１０５に書込まれる書込みデータ１１３を送る際に、それはタイムスタンプ１１４およびシーケンス番号１１５も送る。タイムスタンプ１１４は、ローカルディスクシステムがローカルＨＯＳＴシステム１０２からその要求を受信した時刻を示す。シーケンス番号１１５は書込みデータ１１３のシーケンス番号である。シーケンス番号１１５は、ローカルディスク装置１０４が書込みデータ１１３をリモートディスクシステム１０５に送る時に生成される。ローカルディスクシステム１０４は、シーケンス番号１１４を選定し、それらがシーケンシャルに生成されていることを保証する。

【００２１】データの完全性を実現するために、データの完全性ペアグループ１０９のリモートディスク装置１０７に書込まれるデータの順序は、そのデータの完全性ペアグループの対応するローカルディスク装置１０６に書込まれる同一データの順序と同じである必要がある。データの完全性ペアグループ１０９におけるデータの完全性を保証するために、リモートディスクシステム１０５の中でのタイムスタンプ１１４を比較することが必要である。なぜなら、リモートディスクシステム１０５のうちの１個が７：００のタイムスタンプ１１４を備える書込みデータ１１３を受信したのに対して、リモートディスクシステム１０５の別のものがもっと早いタイムスタンプを備える書込みデータ１１３をすでに受信していることが起こり得るからである。

【００２２】従って、リモートシステム１０１は、複数のスレーブリミットタイムスケジュールプロセス１１６を含む。各リモートディスクシステム１０５は、関係するスレーブリミットタイムスケジュールプロセス１１６を有する。リモートシステムはさらに、単一のマスタリミットタイムスケジュールプロセス１１７を含む。スレーブリミットタイムスケジュールプロセス１１６はそれぞれ、タイムスタンプ１１４の情報をマスタリミットタイムスケジュールプロセス１１７に送る。マスタリミットタイムスケジュールプロセス１１７はその後、リモートディスク装置１０７にデステージングを許可するために最も早い時間を決定する。この最早時間は、タイムリミット１１８として各スレーブタイムリミットスケジュールプロセス１１６に送られる。

【００２３】本発明によれば、ローカルディスクシステム１０４は、データパケットが乱順でリモートシステムに着信した場合でもリモートコピーシステムにおけるデータの完全性を保証するために、書込みデータ１１３の各パケットとともにシーケンス番号１１５を送る。

【００２４】図２を参照すれば、本発明の実施形態において、この図は、すべてのリモートコピーペア１０８が１個のシーケンス番号を有する、ローカルディスクシステムおよびリモートディスクシステムにおけるプロセスを示す。ローカルディスクシステム１０４は、キャッシュメモリ２０２を有するローカルディスク制御装置２００を備える。リモートディスクシステム１０５は、キャッシュメモリ２０３を備えるリモートディスク制御装置２０１を含む。

【００２５】プロセスは、ローカルコンピュータシステム１００において、Ｌ（ローカル）書込みデータ受信プロセス２１０、Ｌ書込みデータ送信プロセス２１１、Ｌ書込みデータ送信完了プロセス２１２およびＬ書込みデータデステージプロセスを含む。リモートコンピュータシステム１０１は、Ｒ（リモート）書込みデータ受信プロセス２１４、Ｒ書込みデータデステージプロセス２１５、マスタリミットタイムスケジュールプロセス１１７およびスレーブリミットタイムスケジュールプロセス１１６を含む。各プロセスは、その各自のローカルディスク制御装置２００またはリモートディスク制御装置２０１において起動する。

【００２６】各リモートコピーペア１０８について、ローカルディスクアドレス２０４、リモートディスクアドレス２０５、データの完全性グループＩＤ２０６およびシーケンスカウンタ２０７を含むリモートコピーペアデータ記憶装置２０３が存在する。ローカルディスクアドレス２０４およびリモートディスクアドレス２０５はそれぞれ、書込み動作のためにリモートコピーペア１０８を構成するローカルディスク装置およびリモートディスク装置の行先アドレスである。データの完全性ペアグループＩＤ２０６は、リモートコピーペアが属するデータの完全性ペアグループ１０９を識別する。図２に例示する通り、各リモートコピーペア１０８は関係するシーケンス番号１１５を有する。シーケンスカウンタ２０７は、リモートディスクに送られる書込みデータ１１３と関係づけられているシーケンス番号１１５を与える。

【００２７】キャッシュメモリ２０２は、各リモートコピーペア１０８について、書込みデータ１１３、リモートディスクアドレス２０８、位置決め情報２０９、シーケンス番号１１５およびタイムスタンプ１１４よりなるデータ構造も含む。このように、ローカルＨＯＳＴ１０２からローカルディスクシステムに送られる書込みデータの各ブロックについて１個の当該データ構造が存在し、それによって、ローカルＨＯＳＴ１０２による典型的な書込み動作のために複数の当該データ構造を生じる。例えば、位置決め情報は、パーソナルコンピュータ

に共通に見られるディスク駆動装置のディスクブロックアドレスである。従来のメインフレームシステムでは、位置決め情報は一般に、シリンダ番号、ヘッド番号およびレコード番号である。

【００２８】ここで図２および５を参照して、Ｌ書込みデータ受信プロセス２１０の処理を説明する。このプロセスは、ローカルＨＯＳＴシステム１０２から書込み要求を受信した時にローカルディスクシステムにおいて実行される。最初に、ローカルディスクシステム１０４はローカルＨＯＳＴシステム１０２から書込み要求を受信する（ステップ５００）。書込み要求は、ローカルディスク装置１０６のアドレス情報および、書込みデータ１１３が書込まれるローカルディスク装置１０６の位置を指定する。次に、ローカルディスクシステム１０４は、書込まれる実際の書込みデータ１１３を受信し、それをキャッシュメモリ２０２にキャッシュする（ステップ５０１）。ローカルディスクシステム１０４は、リモートコピーペア１０８の対応するリモートディスク装置１０７のリモートディスクアドレス２０５をリモートコピーペアデータ記憶装置２０３から取得し、それをリモートディスクアドレス２０８としてキャッシュメモリに格納する。書込み要求において指定された位置は、位置決め情報２０９としてキャッシュメモリに格納される（ステップ５０２）。ローカルディスクシステムはその後、ローカルＨＯＳＴシステムからタイムスタンプを受信し、それをタイムスタンプ１１４としてキャッシュメモリ２０２に格納する（ステップ５０３）。タイムスタンプ１１４は、生成されるタイムスタンプ１１４がローカルディスクシステム１０４の全部に共通である限り、ローカルＨＯＳＴシステム１０４以外によっても生成できる。複数のローカルＨＯＳＴシステム１０２が存在する実施形態では、ＨＯＳＴシステム間で共通のタイムスタンプ値を付与する共有クロックが存在することが前提とされる。最後に、ローカルディスクシステム１０４は、ローカルＨＯＳＴシステム１０２に書込み要求の完了を示す（ステップ５０４）。

【００２９】図２および６を参照して、Ｌ書込みデータ送信プロセス２１１の処理を説明する。このプロセスは、ローカルディスクシステムが書込みデータ１１３をリモートディスクシステム１０５に送る準備ができた時に実行される。図２に示す本発明の実施形態によれば、各リモートコピーペア１０８について１個のＬ書込みデータ送信プロセス２１１が存在する。このプロセスは、Ｌ書込みデータ受信プロセス２１０に関して非同期に実行される。

【００３０】ローカルディスクシステム１０４は、リモートディスクシステムに送られるリモートコピーペア１０８において待機している書込みデータ１１３の全部のうちで時間的に最早関係するタイムスタンプ１１４を有する書込みデータ１１３を選択する（ステップ６００）。ローカルディスクシステムはその後、シーケンスカウンタ２０７の現在値を、選択された書込みデータ１１３（すなわち、その関係するタイムスタンプが時間的に最も早い書込みデータ）に関係づけられるシーケンス番号１１５とみなす。シーケンスカウンタ２０７は増分される（ステップ６０１）。次に、選択された書込みデータ１１３ならびにその関係するタイムスタンプ１１４、シーケンス番号１１５、リモートディスクアドレス情報２０８および位置決め情報２０９は、リモートディスクシステム１０５に送られる（ステップ６０２、図１も参照）。

【００３１】本発明によれば、Ｌ書込みデータ送信プロセス２１１はその後、完了に関するリモートディスクシステム１０５からのいずれの指標も待たずに、次の書込みデータ１１３を処理することに進む。このようにして、高速なデータ転送速度が実現される。しかし、パケットが乱順で着信する可能性がある。それゆえ、ローカルディスクシステム１０４は、リモートディスクシステム１０５に属するいずれかの書込みデータ１１３が、まだ送られていないリモートコピーペア１０８に存在するかどうかを確認する（ステップ６０３）。肯定であれば、処理はステップ６００に継続する。否定であれば、プロセスはしばらく待機し（ステップ６０４）、その後ステップ６０３に継続する。

【００３２】ステップ６０４での待機は、送られる書込みデータがまったく存在しない状況に適応するためである。ステップ６０３で送信する書込みデータがまったく存在しなければ、Ｌ書込みデータ送信プロセス２１１は行うべき処理がまったくない。従って、ステップ６０４は、送信すべき書込みデータがあるかどうかを知るために再び確認する前にプロセスを暫時休止するために使用される。

【００３３】次に図２および７を参照して、Ｌ書込みデータ送信完了プロセス２１２の処理を記明する。このプロセスは、ローカルディスクシステム１０４がリモートディスクシステム１０５への書込みデータ１１３の転送の完了の通知を受信した時に実行される。ローカルディスクシステムは、リモートディスクシステム１０５から書込みデータ１１３の転送の通知を受信する（ステップ７００）。ローカルディスクシステム１０４はその後、対応するタイムスタンプ１１４およびシーケンス番号１１５の値をヌルにする。書込みデータ１１３はすでにリモートディスクシステム１０５に送られているので、これらのフィールドはもはやローカルディスクシステム１０４によって使用されない。

【００３４】リモートディスクシステム１０４への書込みデータ１１３の転送が完了した後、書込みデータ１１３はＬ書込みデータデステージプロセス２１３によってローカルディスク装置１０４に書込まれる。このプロセスは、ローカルディスク装置１０６への書込みデータ１

１３の実際の書込みをもたらし、公知の常法に従って実行される。

【００３５】次に図２および８を参照して、Ｒ書込みデータ受信プロセス２１４を検討する。このプロセスは、ローカルディスクシステム１０４から書込みデータ１１３を受信した時に、リモートディスクシステム１０５において実行される。図２に示す本発明の実施形態によれば、各リモートコピーペア１０８についてＲ書込みデータ受信プロセス２１４が存在する。リモートディスクシステム１０５は、受信された書込みデータ１１３ならびにその関係するタイムスタンプ１１４、シーケンス番号１１５、リモートディスクアドレス情報２０８および位置決め情報２０９をキャッシュメモリ２０３に格納する（ステップ８００）。リモートディスクシステム１０５は、そのシーケンス番号１１５とともに書込みデータ１１３の転送の完了指標をローカルディスクシステムに返送する（ステップ８０１）。

【００３６】キャッシュメモリ２０３は、各リモートコピーペア１０８について、受信された書込みデータ１１３、リモートディスクアドレス２０８、位置決め情報２０９、シーケンス番号１１５およびタイムスタンプ１１４よりなるデータ構造を含む。従って、ローカルＨＯＳＴ１０２から受信された書込みデータ１１３の各ブロックについて１個のそのようなデータ構造が存在し、それによって、複数の当該データ構造を生じる。

【００３７】次に図２および９を参照して、スレーブリミットタイムスケジュールプロセス１１６を説明する。スレーブプロセス１１６は、ある時間実行し、終了する。このプロセスは周期的に起動する。図２に示す本発明の実施形態において、各リモートコピーペア１０８が関係するシーケンス番号１１５を有することを想起されたい。実際、リモートコピーペア１０８において待機している書込みデータ１１３の各ブロックについて、関係するシーケンス番号１１５が存在する。従って、各リモートコピーペア１０８はシーケンス番号１１５のリストを持ち得る。さらに、リモートディスクシステム１０５が、それと関係する複数のリモートコピーペア１０８を有し得ることを想起されたい。従って、各リモートディスクシステム１０５は複数のシーケンス番号のリストを持ち得る。

【００３８】各リモートコピーペア１０８について、スレーブリミットタイムスケジュールプロセス１１６は、そのリモートコピーペアのシーケンス番号のリストを検査する（ステップ９０１）。それは、連続している番号の最長の連続を見つけ、その連続から最大シーケンス番号を返す。例えば、リモートコピーペアが以下のシーケンス番号のリストを含むと仮定する。．．．，１１，１２，１３，１４，１５，１６，１７，１８，１９，２０，２１，２２，２３，２４，２５，２６，２８，２９

【００３９】上記の例では、シーケンス番号２７がまだ受信されていないので、２６以下であるシーケンス番号は連続している。従って、プロセスは、この連続の中で最高値のシーケンス番号である「２６」を選択する。次に、プロセスは、その最高値の（最大の）シーケンス番号と関係するタイムスタンプ１１４を検索する。これが各リモートコピーペアについて繰り返され、タイムスタンプのリストを生じる。

【００４０】このタイムスタンプのリストから、最早タイムスタンプが選択される（ステップ９０１）。各スレーブリミットタイムスケジュールプロセス１１６は、このようにして１個のそうした最早タイムスタンプを生成する。各スレーブプロセスからの最早タイムスタンプはその後、マスタリミットタイムスケジュールプロセス１１７に送られる（ステップ９０２、図１も参照）。各スレーブプロセスはその後、マスタリミットタイムスケジュールプロセス１１７からリミットタイム値１１８を受信するために待ち（ステップ９０３）、値をキャッシュ２０２に格納する（ステップ９０４、図１も参照）。

【００４１】次に図２および１０を参照して、マスタリミットタイムスケジュールプロセス１１７を説明する。このプロセスは、リモートディスクシステムがスレーブプロセス１１６のそれぞれからタイムスタンプ１１４を受信した時に起動する。マスタプロセス１１７は、受信されたタイムスタンプから最早タイムスタンプを選択し（ステップ１０００）、選択されたタイムスタンプをリミットタイム１１８として各スレーブプロセス１１６に送る（ステップ１００１）。

【００４２】次に図２および１１を参照して、Ｒ書込みデータデステージプロセス２１５を説明する。このプロセスは、リモートディスクシステム１０５が、リモートコピーペア１０８に関係する書込みデータ１１３を、そのリモートコピーペア１０８と関係するリモートディスク装置１０７にデステージした時に実行される。リモートディスクシステム１０４は、関係するタイムスタンプ１１４が最も早い候補書込みデータ１１３を選択する（ステップ１１００）。その後それは、選択されたタイムスタンプを、マスタリミットタイムスケジュールプロセス１１７により規定されたリミットタイム１１８と比較する（ステップ１１０１）。選択されたタイムスタンプ１１４がリミットタイム１１８より遅ければ、リモートディスク装置１０５は暫時待機し（ステップ１１０２）、その後、処理をステップ１１００で続ける。選択されたタイムスタンプ１１４がリミットタイム１１８に等しいか、またはそれより早い場合、リモートディスクシステム１０５は、その関係するリモートディスクアドレス２０８および位置決め情報２０９に従って候補書込みデータ１１３をデステージする（すなわち、ディスクに書込む）（ステップ１１０３）。書込みデータならびにその関係するタイムスタンプ、シーケンス番号、リモートディスクアドレス情報２０８および位置決め情報２

０９は、キャッシュメモリ２０３から除去される（ステップ１１０４）。

【００４３】以下の例が役立つであろう。以下を有すると仮定する。

書込みデータＡ、タイムスタンプ１０：００

書込みデータＢ、タイムスタンプ１０：０２

書込みデータＣ、タイムスタンプ１０：０４

リミットタイム：１０：０１

【００４４】この例において、Ｒ書込みデータデスステージプロセスは書込みデータＡを選択し、それは最早タイムスタンプ（１０：００）を有する。次に、Ｒ書込みデータデスステージプロセスは、そのタイムスタンプ（１０：００）がリミットタイム（１０：０１）より早いので、書込みデータＡをリモートディスク装置１０７へデスステージする。書込みデータＡのデスステージングの後、書込みデータＢが最早タイムスタンプを有する。しかし、書込みデータＢのタイムスタンプ（１０：０２）よりリミットタイム（１０：０１）が早いので、書込みデータＢはデステージされることができない。書込みデータＢのデステージングは、リミットタイム１１８がスレーブリミットタイムスケジュールプロセス１１６およびマスタリミットタイムスケジュールプロセス１１７によって１０：０２より遅い時刻に更新された後に可能になる（図９および図１０に示す）。

【００４５】ここで図３を参照して本発明の別の実施形態を説明する。図３のシステム図は、図２に示したものと本質的に同じであるが、以下の相違がある。図３に示す本発明の実施形態では、ローカルディスクシステム１０４と関係づけられている各リモートディスクシステム１０５についてシーケンス番号１１５が存在する。図２では、各リモートコピーペア１０８についてシーケンス番号が存在することを想起されたい。しかし、図３に示す実施形態の場合、ローカルディスクシステム１０４は、少なくとも１個のリモートコピーペア１０８をそのローカルディスクシステムと共有する各リモートディスク１０４システムについて１個のシーケンス番号１１５を有する。従って、図３においてわかる通り、各リモートディスクシステムについて、シーケンスカウンタ３００およびリモートディスクシステムアドレス３０１よりなるデータペアが存在する。従って、ローカルディスクシステムが２個のリモートディスクシステムと関係づけられている場合、ローカルディスクシステムに含まれる２個の当該データペアが存在するはずである。

【００４６】図３に示す実施形態において、Ｌ書込みデータ受信プロセス２１０は、図２に示す実施形態に関連して説明したものと同じである。

【００４７】ここで図２および１２を参照して、図３に示す本発明の実施形態についてＬ書込みデータ送信プロセス２１１を説明する。少なくとも１個のリモートコピーペア１０８をこのローカルディスクシステム１０４と共有する各リモートディスク装置について１個のＬ書込みデータ送信プロセス２１１が存在する。ローカルディスクシステム１０４は、対応するリモートディスクシステム１０５に属する書込みデータ１１３の全部から、タイムスタンプ１１４が時間的に最も早く、リモートディスクシステム１０５にまだ送信されていない書込みデータ１１３を選択する（ステップ１２００）。ローカルディスクシステム１０４はその後、そのリモートディスクシステムに対応するシーケンスカウンタ３００の現在値を、選択された書込みデータ１３３（すなわち、関係するタイムスタンプが時間的に最も早い書込みデータ）と関係するシーケンス番号１１５にコピーする。シーケンスカウンタ３００が増分される（ステップ６０１）。次に、選択された書込みデータ１１３ならびにその関係するタイムスタンプ１１４、シーケンス番号１１５、リモートディスクアドレス情報２０８および位置決め情報２０９が、リモートディスクシステム１０５に送信される（ステップ６０２）。プロセスはその後、図６の説明に従って継続する。

【００４８】図３に示す本発明の実施形態において、Ｌ書込み送信完了プロセス２１２は、図２に示す本発明の実施形態に関連して説明したものと同じである。

【００４９】図３に示す実施形態において、リモートデディスクシステム１０５は、リモートコピーペア１０８をリモートディスクシステム１０５と共有するすべてのローカルディスクシステム１０４について、シーケンスカウンタ３００およびローカルディスクシステムアドレス３０２よりなるデータペアを含む。リモートディスクシステムと関係する各ローカルディスクシステムについて１個の当該データペアが存在する。

【００５０】リモートコピーペア１０８をローカルディスクシステム１０５と共有するすべてのローカルディスクシステムについてＲ書込みデータ受信プロセス２１４が存在する。図３の実施形態において、Ｒ書込みデータ受信プロセス２１４は、図２に示す実施形態に関連して説明したものと同じである。問様に、マスタリミットタイムスケジュールプロセス１１７およびＲ書込みデータデスステージプロセス２１５も図２に示す本発明の実施形態の説明と同じである。

【００５１】ここで図１３を参照して、図３に示す本発明の実施形態に従ったスレーブリミットタイムスケジュールプロセス１１６を検討する。各リモートディスクシステム１０５について１個のスレーブプロセスが存在する。このプロセスは、以下の相違はあるが、図２の実施形態に関してと本質的に同じである。

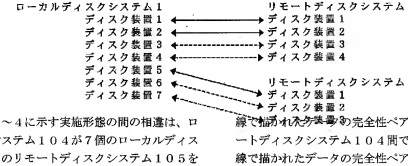
【００５２】少なくとも１個のリモートコピーペア１０８を共有する各ローカルディスクシステム１０４について、スレーブリミットタイムスケジュールプロセス１１６は、そのローカルディスクシステム１０４から受信されたシーケンス番号のリストを検査する（ステップ１３

００）。プロセスは、連続であるシーケンス番号１１４の最大シーケンス番号を見つけ、その最大シーケンス番号に対応するタイムスタンプ１１５を返す。これは、タイムスタンプのリストを生成するために、少なくとも１個のリモートコピーペア１０８をリモートディスクシステムと共有する各ローカルディスクシステムについて繰り返される。このタイムスタンプのリスト１１から、最早タイムスタンプが選択され（ステップ９０１）、マスタタイムリミットスケジュールプロセス１１７に送達される（ステップ９０２）。このようにして、各スレーブリミットタイムスケジュールプロセス１１６は、各リモートディスクシステム１０５について１個のそうした最早タイムスタンプを探索する。プロセスはその後、マスタリミットタイムスケジュールプロセス１１７からリミットタイム値１１８を受信するために待機し（ステップ９０３）、値をキャッシュ２０２に格納する（ステップ９０４）。

【００５３】次に図４を参照して、本発明の別の実施形態を説明する。図４のシステム図は、以下の相違はあるが、図２に関して示したものと本質的に同じである。図４に示す本発明の実施形態において、ローカルディスクシステムは、そのローカルディスクシステム１０４と共通してデータの完全性ペアグループ１０９を有する各リモートディスクシステム１０５について１個のシーケンス番号１１５を有する。例えば、ローカルディスクシステムＸが、関係するデータの完全性ペアグループＺを有するとする。データの完全性ペアグループはさらに、５個のリモートコピーペアグループＲＣＰ１～ＲＣＰ５よりなる。さらに、リモートコピーペアグループＲＣＰ１およびＲＣＰ２はリモートディスクシステムＡと関係し、リモートコピーペアＲＣＰ３はリモートディスクシステムＢと関係し、ＲＣＰ４およびＲＣＰ５はリモートディスクシステムＣと関係するとみなす。この特定の例において、ローカルディスクシステムＸは、データの完全性ペアグループＺと関係する３個のシーケンス番号を有する。

【００５４】

【表１】




【００５５】図２～４に示す実施形態の間の相違は、ローカルディスクシステム１０４が７個のローカルディスク装置および２個のリモートディスクシステム１０５を有する、上記の例示的な機器構成に示されている。両矢印はリモートコピーペアを示す。例えば、ローカルディスク装置５およびリモートディスク装置１は、リモートコピーペア１０８を規定する。さらに、実線で描かれたリモートコピーペアは一体で１個のデータの完全性グループを規定するのに対し、破線で描かれたリモートコピーペアは一体で第２のデータの完全性グループを規定する。

【００５６】従って、図２に示された本発明の実施形態によれば、ローカルディスクシステム１０４はそのディスク装置（合計７個のシーケンス番号）のそれぞれについて関係するシーケンス番号１１４を有するであろう。図４に示す実施形態によれば、ローカルディスクシステム１０４は、ローカルディスクシステム１０５と共通してリモートコピーペアを有する各リモートディスクシステム１０５について１個ずつ、２個の関係するシーケンス番号を有する。図４に示す実施形態によれば、ローカルディスクシステム１０４は、データの完全性ペアグループ１０９をローカルディスクシステム１０４と共有する各リモートディスクシステム１０５について１個ずつ、４個の関係するシーケンス番号１１４を有する。実線で描かれたデータの完全性ペアグループは２個のリモートディスクシステム１０４間で共有され、同様に、破線で描かれたデータの完全性ペアグループは２個のリモートディスクシステム１０４間で共有される。

【００５７】さらに図４について続ければ、図示の通り、データの完全性ペアグループ１０９によってローカルディスクシステム１０４と関係づけられている各リモートディスクシステム１０５について、データの完全性ペアグループ情報データ構造４００がローカルディスクシステム１０４に存在する。データ構造４００は、データの完全性グループＩＤ４０１および、リモートディスクシステムアドレス４０３およびシーケンスカウンタ４０２よりなる１個以上のデータペアを含む。データの完全性ペアグループ１０９の各リモートディスクシステム１０５について１個の当該データペアが存在する。リモートディスクシステム１０５は、データの完全性ペアグループ１０８によってリモートディスクシステム１０５と関係づけられている各ローカルディスクシステム１０４について類似のデータ構造を備える。

【００５８】図４に示す本発明の実施形態において、Ｌ書込み送信完了プロセス２１０およびＬ書込み送信完了プロセス２１２は、図２に示す本発明の実施形態に従って動作する。

【００５９】次に図１４を参照して、図４に示す本発明

の実施形態のＬ書込みデータ送信プロセス２１１を説明する。少なくとも１個のデータの完全性ペアグループ１０９をそのローカルディスクシステム１０４と共有する各リモートディスク装置についてＬ書込みデータ送信プロセス２１１が存在する。ローカルディスクシステム１０４は、対応するデータの完全性ペアグループ１０９に属する書込みデータ１１３の全部から、タイムスタンプ１１４が時間的に最も早く、リモートディスクシステム１０５にまだ送信されていない書込みデータ１１３を選択する（ステップ１４００）。ローカルディスクシステムはその後、ターゲットリモートディスクシステムに対応するシーケンスカウンタ４０３の現在値を、選択された書込みデータ１１３（すなわち、関係するタイムスタンプが時間的に最も早い書込みデータ）と関係するシーケンス番号１１５にコピーする。シーケンスカウンタ４０３が増分される（ステップ６０１）。次に、選択された書込みデータ１１３ならびにその関係するタイムスタンプ１１４、シーケンス番号１１５、リモートディスクアドレス情報２０８および位置決め情報２０９が、リモートディスクシステムに送られる（ステップ６０２）。プロセスはその後、図６に関する説明に従って継続する。

【００６０】図４に示す本発明の実施形態において、Ｒ書込みデータ受信プロセス２１４は、そのリモートディスクシステム１０５と関係するデータの完全性ペアグループ１０９を共有する各ローカルディスク装置１０４と関係づけられている。処理は図８に示すフローチャートに従って進行する。

【００６１】マスタリミットタイムスケジュールプロセス１１７およびＲ書込みデータデステージプロセス２１５は、図２に示す本発明の実施形態に関する説明と同様にして動作する。

【００６２】次に図１５を参照して、図４に示す本発明の実施形態に従ったスレーブリミットタイムスケジュールプロセス１１６を検討する。各リモートディスクシステム１０５について１個のスレーブプロセスが存在する。このプロセスは、以下の相違はあるが、図２の実施形態に関して上述のに同じである。

【００６３】データの完全性ペアグループによってリモートディスクシステムと関係づけられている各ローカルディスクシステム１０４について、スレーブリミットタイムスケジュールプロセス１１６は、そのローカルディスクシステムから受信されたシーケンス番号のリストを検査する（ステップ１５００）。プロセスは、連続であるシーケンス番号１１４の最大シーケンス番号を見つけ、最大シーケンス番号に対応するタイムスタンプ１１５を返す。これは、タイムスタンプのリストを生成するために、データの完全性ペアグループ１０９によってリモートディスクシステムと関係する各ローカルディスクシステム１０４について繰り返される。タイムスタンプのこのリストから、最早タイムスタンプが選択され（ステップ９０１）、マスタタイムリミットスケジュールプロセス１１７に送達される（ステップ９０２）。このようにして、各スレーブリミットタイムスケジュールプロセス１１６は、各リモートディスクシステム１０５について１個のそうした最早タイムスタンプを探索する。プロセスはその後、マスタリミットタイムスケジュールプロセス１１７からリミットタイム値１１８を受信するために待機し（ステップ９０３）、値をキャッシュ２０２に格納する（ステップ９０４）。

【００６４】ここで、リモートサイトに磁気テープ（ＭＴ）記憶装置システムが配備された状況における本発明の実施形態の検討のために図１６に話を移す。ローカルコンピュータシステム１６００は、少なくともローカルＨＯＳＴシステム１６０２および、ローカルディスクシステムといった少なくともローカル記憶装置システム１６０４を含む。ローカルコンピュータシステムはスイッチシステム１６０６を含み得る。少なくともリモートＭＴシステム１６０５を含むリモートコンピュータシステム１６０１が存在する。リモートＨＯＳＴシステム１６０３は、必ずしもリモートコンピュータシステム１６０１に存在するわけではない。

【００６５】ローカルＨＯＳＴシステム１６０１、ローカルディスクシステム１６０４およびスイッチシステム１６０６は、通信路１６１３によってリモートディスクシステム１６０５と接続される。通信路１６１３は、その送信データパケットが必ずしもそれらが送られた順序で着信するわけではないことを特徴とするネットワークである。例えば、ＩＰベースのネットワークはこの振舞いを呈する。本発明のこの実施形態において、読取り／書込みデータの単位をブロックと称する。各ブロックは、自身を識別するために使用されるブロックＩＤを有する。

【００６６】ローカル記憶装置システム１６０４は、少なくとも１個のローカル記憶装置１６０７および、キャッシュメモリ１６１１を備える記憶制御装置１６０９を有する。リモートＭＴシステム１６０５は、少なくとも１個のリモートＭＴ装置１６０８および、キャッシュメモリ１６１１を備えるＭＴ制御装置１６１０を含む。ローカルＨＯＳＴシステム１６０２はメインメモリ１６１２を有する。スイッチシステム１６０６はキャッシュメモリ１６１１も備える。

【００６７】図１６に示す本発明の実施形態によれば、リモートＭＴ装置１６０８への読出し要求および書込み要求はそれぞれ、読み出されるデータ１６２４または書込まれるデータ１６２５に伴うシーケンス番号１６１４を含む。

【００６８】図１６はまた、ローカルＨＯＳＴシステム１６０２、ローカルディスクシステム１６０４、リモートＭＴシステム１６０５およびスイッチシステム１６０

６のプロセスも示している。読出し要求発行プロセス１６１５および書込み要求発行プロセス１６１６は、ローカルＨＯＳＴシステム１６０２、ローカルディスクシステム１６０４およびスイッチシステム１６０６において提供される。リモートＭＴシステム１６０５は、読出し要求受信プロセス１６１７および書込み要求受信プロセス１６１８を含む。

【００６９】読出し要求発行プロセス１６１５の処理は図１７に概説されている。このフローは、ローカルＨＯＳＴシステム１６０２、ローカル記憶装置システム１６０４およびスイッチシステム１６０６の間で共通である。ローカル記憶制御装置１６０９のキャッシュメモリ１６１１に（またはメインメモリ１６１２に）コピーされるシーケンスカウンタ１６２３が、読取り／書込み要求が発行された時に、シーケンス番号１６１４の値を付与する（ステップ１７００）。プロセスはその後、読出し要求をリモートＭＴシステム１６０５に発行する（ステップ１７０１）。プロセスはその後、シーケンスカウンタ１６２３を増分する（ステップ１７０２）。

【００７０】次に（ステップ１７０３）、読出し要求発行プロセスは、発行された要求の数を確認する。その数が値ｍ未満であれば、処理は、次の要求を発行するためにステップ１７００に継続する。ｍの値は一般に＞２であり、好ましくは経験的に決定される。その目的は、現在の要求の完了前に次の要求を送信することによってより良好な性能を得ることである。

【００７１】数がｍ未満でなければ、プロセスは、リモートＭＴシステム１６０５からのデータ転送通知を待つ（ステップ１７０４）。読出し要求発行プロセス１６１５がリモートＭＴシステム１６０５からのデータ転送通知を受信すると、それは読出しデータ１６２４およびシーケンス番号１６１４を受信し、シーケンスカウンタ１６２３に従ってそれをキャッシュメモリ１６１１に（またはメインメモリ１６１２に）格納する（ステップ１７０５）。次に、読出し要求発行プロセス１６１５は、リモートＭＴシステム１６０５から全部のデータ転送通知を受信したかどうかを確認する（ステップ１７０６）。否定であれば、プロセスは、データ転送通知を待つためにステップ１７０３に継続する。

【００７２】書込み要求発行プロセス１６１６の処理は、図１８に示すフローチャートに概説されている。このフローは、ローカルＨＯＳＴシステム１６０２、ローカル記憶装置システム１６０４およびスイッチシステム１６０６間で共通である。書込み要求発行プロセスは、シーケンス番号１６１４に対するシーケンスカウンタ１６２３の内容をキャッシュメモリ１６１１に（またはメインメモリ１６１２に）コピーし（ステップ１８００）、シーケンス番号１６１４および書込みデータ１６２５を伴う書込み要求をリモートＭＴシステム１６０５に発行する（ステップ１８０１）。その後、シーケンスカウンタ１６２３を増分する（ステップ１８０２）。ステップ１８０３において、書込み要求発行プロセス１６１６は、発行された要求の数を確認する。その数が値ｍ未満であれば、次の要求を発行するためにステップ１８００にジャンプする。そうでなければ、プロセスはリモートＭＴシステムからのデータ転送通知を待つ（ステップ１８０４）。書込み要求発行プロセス１６１６がリモートＭＴシステム１６０５から通知を受信すると（ステップ１８０５）、プロセスはリモートＭＴシステム１６０５から全部のデータ転送通知を受信したかどうかを確認する（ステップ１８０６）。否定であれば、データ転送通知を待つためにステップ１８０３にジャンプする。

【００７３】リモートＭＴシステム１６０５の読出し要求受信プロセス１６１７のフローは、図１９に示す。リモートＭＴシステム１６０５は、ＭＴ制御装置１６１０のキャッシュメモリ１６１１に読出しデータ１６２４を有する。１群の読出しデータ１６２４が、１個のリモートＭＴ装置１６０８から読み出され、ＭＴ待ち行列１６２６に格納される。各待ち行列エントリは、読出しデータ１６２４およびその対応するブロックＩＤ情報１６２７よりなるデータペアである。読出しデータ１６２４およびそのブロックＩＤ情報１６２７は、ブロックＩＤ情報１６２７に従って待ち行列に挿入される。従って、ＭＴ待ち行列１６２６はブロックＩＤ情報１６２７によって順序づけられる。この順序づけは、データブロックが要求側ローカルＨＯＳＴ１６０１に返送されるべき順序を示している。従って、一連の読出し要求は、適切な順番で受信されていれば、単に各データブロックを、それらがＭＴ待ち行列１６２６に存在する順序で送ることによって満たされるであろう。

【００７４】待ち行列１６２６は、二重連係データ構造である。それゆえ、各待ち行列エントリはさらに、次のブロックＩＤ情報１６２７を含む待ち行列エントリをポイントする正方向ポインタ１６３０および、以前のブロックＩＤ情報１６２７を含む待ち行列エントリをポイントする逆方向ポインタ１６３１を含む。ヘッドポインタ１６２８がＭＴ待ち行列１６２６の先頭をポイントし、テールポインタ１６２９がＭＴ待ち行列１６２６の末尾をポイントする。次のシーケンス番号カウンタ１６３２は、次の読出し要求によりアクセスされるべき読出しデータ１６２４に対応するシーケンス番号１６１４を含む。それゆえ、カウンタ１６３２の現在値は、ＭＴ待ち行列の先頭のデータブロックに対応する。

【００７５】動作時、読出し要求受信プロセス１６１７は、ローカルＨＯＳＴ１６０１からシーケンス番号１６１４とともに読出し要求を受信する（ステップ１９００）。読出し要求受信プロセスは、受信されたシーケンス番号１６１４を次のシーケンス番号カウンタ１６３２と比較する（ステップ１９０１）。それが等しい場合、プロセスは、受信されたシーケンス番号１６１４およ

び、（ヘッドポインタ１６２８によってポイントされる）ＭＴ待ち行列１６２６の先頭の読出しデータ１６２４を、受信されたシーケンス番号１６１４とともに、読出し要求発行プロセス１６１５へ送る（ステップ１９０２）。シーケンス番号カウンタ１６３２はその後更新され、ＭＴ待ち行列の次の読出しデータ１６２４のブロックＩＤ情報１６２７を参照する（ステップ１９０３）。プロセスはその後ステップ１９０７に継続する。

【００７６】受信されたシーケンス番号が次のシーケンス番号カウンタ１６３２と等しくない場合、それは、乱順の読出し要求が受信されたことを意味する。ステップ１９０４において、読出し要求受信プロセス１６１７は、要求されている読出しデータのブロックＩＤ情報１６２７を計算する。詳細には、読出し要求受信プロセスは、次のシーケンス番号カウンタ１６３２と受信されたシーケンス番号１６１４との間の差Ｄを取得し、その差ＤをＭＴ待ち行列１６２６の先頭の読出しデータ１６２４のブロックＩＤ情報Ｂに加算する。すなわち、Ｂ＋Ｄは見つけるべきブロックＩＤ情報１６２７である。ステップ１９０５において、読出し要求受信プロセスは、ブロックＩＤ情報１６２７がＢ＋Ｄである待ち行列エントリを探索する。待ち行列１６２６のそのエントリにある読出しデータ１６２４は、受信されたシーケンス番号１６１４とともに、読出し要求発行プロセス１６１５に送られる（ステップ１９０６）。処理はステップ１９０７に続く。

【００７７】ステップ１９０７において、読出し要求受信プロセス１６１７は、送信された読出しデータ１６２４に対応する待ち行列エントリを削除する。このように、シーケンス番号１６１４を使用することによって、リモートＭＴシステム１６０５は、たとえ読出し要求が通信路１６１３の性質のため乱順になった場合でも間違いなくアクセスされた読出しデータ１６２４を認識することができる。図１７に戻って言えば、受信データは、受信されたシーケンス番号１６１４に従って組み立てられる。従って、リモートシステムにおける読出し要求が乱順になった場合、シーケンス番号は、それらが正しい順序で満たされるように保証する。同様に、ローカルシステムにおいて受信データが乱順になった場合も、それはシーケンス番号１６１４により正しく組み立てられる。

【００７８】リモートＭＴシステム１６０５における書込み要求受信プロセス１６１８のフローは、図２０に示す。書込まれるデータのＭＴ待ち行列１６２６の構造は、読出しデータ１６２４のそれと同じである。書込みデータ１６２５の場合、ＭＴ装置１６０５の次のシーケンス番号カウンタ１６３２は、テールポインタ１６２９によってポイントされる書込みデータのシーケンス番号１６１４よりも１だけ大きい。

【００７９】リモートコンピュータにおける書込み要求受信プロセス１６１８は、シーケンス番号１６１４および書込みデータ１６２５を伴う書込み要求を受信し、書込みデータをキャッシュメモリ１６１１に格納する（ステップ２０００）。次に、書込み要求受信プロセス１６１８は、シーケンス番号１６１４が次のシーケンス番号カウンタ１６３２と等しいかまたはそれより大きいかどうかを確認する（ステップ２００１）。等しければ、それは対応する書込みＭＴ待ち行列１６２６の末尾の受信された書込みデータ１６２５を挿入し、従ってＭＴ待ち行列の情報を更新する（ステップ２００２）。プロセスはその後、受信された書込みデータ１６２５のブロックＩＤを計算し、それをブロックＩＤ情報１６２７に格納する（ステップ２００３）。その後、Ｄ２（シーケンス番号１６１４と次のシーケンス番号カウンタ１６３２との差）＋１をシーケンス番号カウンタ１６３２に加算し（ステップ２００４）、ステップ２００９にジャンプする。

【００８０】受信されたシーケンス番号が次のシーケンス番号カウンタ１６３２より小さい場合、プロセスは、次のシーケンス番号カウンタ１６３２と受信されたシーケンス番号１６１４との差を計算する（ステップ２００５）。プロセスは、書込みデータ１６２４を挿入するためのＭＴ待ち行列１６２６の位置を見つけ、従ってＭＴ待ち行列の情報を更新する（ステップ２００７）。その後プロセスは、受信された書込みデータ１６２５のブロックＩＤを計算し、それをブロックＩＤ情報１６２７に格納する（ステップ２００８）。次に、書込み要求受信プロセスは書込み要求の完了を通知し、受信されたシーケンス番号１６１４を書込み要求発行プロセス１６１６に返送する（ステップ２００９）。このようにして、シーケンス番号１６１４を使用することによって、リモートＭＴシステム１６０５はやはり、たとえ書込み要求が通信路１６１３のために乱順になった場合でも間違いなく書込みデータ１６２５のブロックＩＤを認識することができる。

【００８１】図２１は、ＭＴ記憶装置システムに関する本発明の別の実施形態を示す。本発明のこの実施形態は、図１６に示す実施形態と本質的に同じである。相違は、乱順の発生を検出し、当該発生を訂正するために、図１６において使用されたシーケンス番号１６１４の代わりに次のブロックＩＤ情報１７００、１７０１を使用することである。

【００８２】

【発明の効果】本発明によれば、信頼できるＩＰベースのデータ回復システムを提供することができる。

【図面の簡単な説明】

【図１】本発明の一般化したシステム図である。

【図２】シーケンス番号がリモートコピーペアごとに割り当てられる本発明の１実施形態の一般化したブロック図である。

【図３】シーケンス番号がリモートディスクシステムごとに維持される本発明の別の実施形態の一般化したブロック図である。
【図４】シーケンス番号がデータの完全性ペアグループごとに管理される本発明の別の実施形態の一般化したブロック図である。
【図５】本発明の１実施形態に従ったプロセスフローを概説する。
【図６】本発明の１実施形態に従ったプロセスフローを概説する。
【図７】本発明の１実施形態に従ったプロセスフローを概説する。
【図８】本発明の１実施形態に従ったプロセスフローを概説する。
【図９】本発明の１実施形態に従ったプロセスフローを概説する。
【図１０】本発明の１実施形態に従ったプロセスフローを概説する。
【図１１】本発明の１実施形態に従ったプロセスフローを概説する。
【図１２】図６に示すプロセスフローの代替実施形態を示す。
【図１３】図９に示すプロセスフローの代替実施形態を示す。
【図１４】図６に示すプロセスフローの別の代替実施形態を示す。
【図１５】図９に示すプロセスフローの別の代替実施形態を示す。
【図１６】本発明の磁気テープ記憶装置システムの実施形態を示す。
【図１７】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。
【図１８】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。
【図１９】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。
【図２０】磁気テープ記憶装置システムによる本発明の実施形態におけるプロセスフローを示す。
【図２１】本発明の別の磁気テープ実施形態である。
【符号の説明】
１００…ローカルコンピュータシステム、１０１…リモートコンピュータシステム、１０２…ローカルＨＯＳＴシステム、１０３…リモートＨＯＳＴシステム、１０４…ローカルディスクシステム、１０５…リモートディスクシステム、１０６…ローカルディスク装置、１０７…リモートディスク装置、１０８…リモートコピーペア、１０９…データの完全性ペアグループ、２００…ローカルディスク制御装置、２０１…リモートディスク制御装置、２０２…キャッシュメモリ、４０２…シーケンスカウンタ、１６０５…リモートＭＴシステム、１６０６…スイッチシステム、１６０８…リモートＭＴ装置、１６１０…ＭＴ制御装置、１６１２…メインメモリ。

# Storage System Connected to a Data Network With Data Integrity

## BACKGROUND OF THE INVENTION

The present invention relates generally to data storage systems and more particularly to maintaining data integrity of data storage systems in a data network environment.

Conventionally, data processing systems have access to their associated data storage systems over a high speed, high reliability data bus. However, opportunities become available as the widespread use of network communications continues to expand. IP (Internet Protocol) provides the basic packet delivery service on which TCP/IP (transport control protocol/IP) networks are built. IP is a well defined protocol and is therefore a natural candidate for providing the transport/networking layer for network-based data storage access, where server systems exchange data with storage systems and storage systems exchange data with other storage systems using IP.

The nature of IP, however, presents some unique problems in the area of data storage access systems. First, IP is a connectionless protocol. This means that IP does not exchange control information to establish an end-to-end connection prior to transmitting data. IP contains no error detection and recovery mechanism. Thus, while IP can be relied on to deliver data to a connected network, there is no mechanism to ensure the data was correctly received or that the data is received in the order that it was sent. IP relies on higher layer protocols to establish the connection if connection-oriented service is desired.

In a data storage system where dual remote copy capability is needed, IP-based transmission presents a problem. A remote copy function provides a real time copy of a primary data store at a remote site with the goal of realizing disaster recovery in the primary data store. It is important to guarantee data integrity in order that this function serves its purpose. There are two types of remote copy: synchronous and asynchronous.

In a synchronous type remote copy, a write request by a local HOST to its associated local disk system does not complete until after the written data is transferred from the local disk system to a remote disk system. Thus, in the case of synchronous type copy, it is easy to ensure data integrity between the local and the remote disk system.

In an asynchronous type remote copy, a write request by the local HOST completes before the local disk completes its transfer to the remote disk. As the name

implies, control is returned to the local HOST irrespective of whether the transfer operation from the local disk to the remote disk completes. Data integrity during an asynchronous type copy operation, therefore, relies on a correct arrival order of data at the remote disk system so that data on the remote disk is written in the same order as on the local disk.

To achieve this, the local disk system includes a time stamp with the data that is sent to the remote disk. Data at the remote is written according to the order of the time stamp. Thus, for example, when the remote disk receives data with a time stamp 7:00, it has already received all data whose time stamps precede 7:00.

However, in an IP-based network, when packets can arrive out of sequence, a data packet having a time stamp of 7:00 may or may not be preceded by data packets having an earlier time stamp. Consequently, it is difficult to ensure data integrity at a remote disk system when the transmission protocol is based on connectionless transport model such as the IP.

Another problem arises when IP is used with magnetic tape systems. Read and write operations to magnetic tape is sequential and so the addressing is fixed. Thus, in the case where a data packet arrives at the remote site out of sequence, the data will be written to tape in incorrect order. A subsequent recovery operation from tape to restore a crashed storage system would result in corrupted data.

There is a need to provide a reliable IP-based data recovery system.

## SUMMARY OF THE INVENTION

A data storage system in accordance with the invention comprises a local storage component and a remote storage component. Data to be written to the local storage component is sent in a data packet to the remote storage component over a data network.

The data packet includes a copy of the data to be written at the local storage component, a time stamp, and a sequence number. Plural such data packets are received at the remote storage component. The data packets are selected for writing at the remote storage component based on the sequence numbers and the time stamps associated with each data packet.

In one embodiment of the invention, the local and remote storage components are configured as plural local and remote disk units, respectively. Each local

disk unit is associated with a remote disk unit. Each such pairing is called a remote copy pair. In this embodiment, each remote copy unit has an associated sequence number.

In another embodiment of the invention, the local disk units are grouped into local disk systems. Similarly, each remote disk unit is grouped into remote disk systems. In this embodiment of the invention, there is a sequence number for each pair of local and remote disk systems which have at least one common remote copy pair.

In another embodiment of the invention, the remote copy pairs are grouped into data integrity pair groups. In this embodiment, there is a sequence number for each pair of local and remote disk systems which have in common at least one data integrity pair group.

## BRIEF DESCRIPTION OF THE DRAWINGS

The teachings of the present invention can be readily understood by considering the accompanying detailed description in conjunction with the following drawings:

Fig. 1 is a generalized system diagram of the present invention;

Fig. 2 is a generalized block diagram of one embodiment of the invention in which sequence numbers are allocated on a per remote copy pair basis;

Fig. 3 is a generalized block diagram of another embodiment of the invention in which sequence numbers are maintained on a per remote disk system basis;

Fig. 4 is a generalized block diagram of another embodiment of the invention in which sequence numbers are managed on a per data integrity pair group basis;

Figs. 5 – 11 outline process flows in accordance with one embodiment of the invention;

Fig. 12 shows an alternate embodiment of the process flow shown in Fig. 6;

Fig. 13 shows an alternate embodiment of the process flow shown in Fig. 9;

Fig. 14 shows another alternate embodiment of the process flow shown in Fig. 6;

Fig. 15 shows another alternate embodiment of the process flow shown in Fig. 9;

Fig. 16 shows a magnetic tape storage system embodiment of the present invention;

Figs. 17 – 20 show process flows in an embodiment of the invention in a magnetic tape storage system; and

Fig. 21 is another magnetic tape embodiment of the invention.

## DESCRIPTION OF THE SPECIFIC EMBODIMENTS

Referring to Fig. 1, a computer system in accordance with one embodiment of the invention is shown. A local computer system 100 includes at least a local HOST system 102 and at least a local disk system 104. The remote computer system 101 includes at least a remote disk system 105. A remote HOST system 103 is not always necessary for the remote computer system 101. Each local disk system 104 is connected to a remote disk system 105 through a communication path 112. The communication path 112 is a network which is characterized in that transmitted data packets do not necessarily arrive in the order in which they were sent. An IP-based network exhibits this behavior. In general, a connectionless network exhibits this behavior. As an example, a wide area network (WAN) can be based on IP. The invention, however, is not limited to WAN's.

The local disk system 104 includes at least a local disk unit 106 which has an associated real time copy maintained in the remote disk system 105. The remote disk system 105 includes at least a remote disk unit 107 which contains a real time copy of a local disk unit 106. The pair comprising a local disk unit 106 and a remote disk unit 107 is called a remote copy pair 108. A group of remote copy pairs 108, among which data integrity must be guaranteed, is called a data integrity pair group 109. A group of local disk units 106 which belong to a data integrity pair group 109 is called a data integrity local disk group 110. Similarly at the remote system, a group of remote disk units 107 which belong to one data integrity pair group 109 is called a data integrity remote disk group 111.

The data integrity local disk group 110 may comprise local disk units 106 from a single local disk system 104, or from two or more local disk systems 104. Consequently, the constituent local disk units 106 of a local disk system 104 may found in one or more data integrity pair groups 109. The data integrity remote disk unit group 111 may comprise remote disk units 107 which belong to one or more remote disk

systems 105. Consequently, a remote disk system 105 may have remote disk units 107 which belong to different data integrity pair groups 109.

During the course of operation of the local HOST system 102, data will need to be written to the local disk system 104. The local HOST will transfer "write data" 113 to be stored on the local disk system 104. The local disk system 104 also sends the write data 113 to a remote disk system 105 for data recovery. In accordance with the invention, when the local disk system 104 sends the write data 113 to be written to the remote disk system 105, it also sends time stamp 114 and a sequence number 115. The time stamp 114 shows the time when the local disk system received the request from the local HOST system 102. The sequence number 115 is the sequence number of the write data 113. The sequence number 115 is generated when a local disk unit 104 sends write data 113 to a remote disk system 105. The local disk system 104 selects the sequence number 114 and ensures that they are sequentially generated.

To realize data integrity, the order of data written on the remote disk units 107 in a data integrity pair group 109 needs to be same as the order of the same data written on the corresponding local disk units 106 of that data integrity pair group. To guarantee the data integrity in the data integrity pair group 109, it is necessary to compare the time stamps 114 from among the remote disk systems 105 because it is possible that one of the remote disk systems 105 will have received write data 113 with a time stamp 114 of 7:00, while another of the remote disk systems 105 has yet to receive write data 113 with an earlier time stamp.

Thus, the remote system 101 includes plural slave limit time schedule processes 116. Each remote disk system 105 has an associated slave limit time schedule process 116. The remote system further includes a single master limit time schedule process 117. The slave limit time schedule processes 116 each send information for time stamp 114 to the master limit time schedule process 117. The master limit time schedule process 117 then decides the earliest time to permit de-staging to a remote disk unit 107. This earliest time is sent as a time limit 118 to each slave time limit scheduling process 116.

In accordance with the invention, the local disk system 104 sends a sequence number 115 with each packet of write data 113 to ensure data integrity in the remote copy system, even in the event that the data packets arriving at the remote system arrive out of order.

Referring to Fig. 2, in an embodiment of the invention, the figure shows the processes in the local disk system and the remote disk system where every remote copy pair 108 has one sequence number. The local disk system 104 includes a local disk control unit 200 having a cache memory 202. The remote disk system 105 includes a remote disk control unit 201 with a cache memory 203.

The processes include an L-write data receive process 210, an L-write data send process 211, an L-write data send completion process 212, and an L-write data destage process on the local computer system 100. The remote computer system 101 includes an R-write data receive process 214, an R-write data destage process 215, and the master and slave limit time scheduling processes, 117, 116 respectively. Each process is activated in its respective local disk control unit 200 or remote disk control unit 201.

For each remote copy pair 108, there is a remote copy pair data store 203 which includes a local disk address 204, a remote disk address 205, a data integrity group id 206, and a sequence counter 207. The local disk address 204 and the remote disk address 205 are the destination addresses respectively of the local disk unit and remote disk unit which comprise the remote copy pair 108, for a write operation. The data integrity pair group id 206 identifies to which data integrity pair group 109 the remote copy pair belongs. As shown in Fig. 2, each remote copy pair 108 has an associated sequence number 115. The sequence counter 207 provides the sequence number 115 that is associated with the write data 113 that is sent to the remote disk.

The cache memory 202 also contains, for each remote copy pair 108, a data structure comprising write data 113, a remote disk address 208, positioning information 209, a sequence number 115, and a time stamp 114. Thus, there is one such data structure for each block of write data that is sent to the local disk system from the local HOST 102, thereby resulting in plural such data structures for a typical write operation by local HOST 102. The positioning information is, for example, the disk block address on disk drives commonly found in personal computers. In conventional mainframe systems, the positioning information is typically the cylinder number, head number, and record number.

Referring now to Figs. 2 and 5, the processing of the L-write data receive process 210 will be described. This process is executed in the local disk system when it receives a write request from the local HOST system 102. First, the local disk system 104 receives a write request from the local HOST system 102 (step 500). A write request specifies the address information of the local disk unit 106 and the position on the local

disk unit 106 to which the write data 113 is written. Next, the local disk system 104 receives the actual write data 113 to be written and caches it in cache memory 202 (step 501). The local disk system 104 obtains the remote disk address 205 of the corresponding remote disk unit 107 in the remote copy pair 108 from the remote copy pair data store 203 and stores it in the cache memory as remote disk address 208. The position specified in the write request is stored in the cache memory as position information 209 (step 502). The local disk system then receives a time stamp from the local HOST system and stores it into the cache memory 202 as time stamp 114 (step 503). The time stamp 114 can be generated other than by the local HOST system 104 , so long as the time stamp 114 that is produced is common to all of the local disk systems 104. In an embodiment where there are more than one local HOST systems 102, a shared clock is assumed to exist which provides a common time stamp value amongst the HOST systems. Finally, the local disk system 104 indicates to the local HOST system 102 the completion of the write request (step 504).

Referring to Figs. 2 and 6, the processing of the L-write data send process 211 will be described. This process is executed when the local disk system is ready to send write data 113 to the remote disk system 105. In accordance with the embodiment of the invention shown in Fig. 2, there is an L-write data send process 211 for each remote copy pair 108. This process is executed asynchronously relative to the L-write data receive process 210.

The local disk system 104 selects the write data 113 whose associated time stamp 114 is the earliest in time from among all of the write data 113 that are waiting in the remote copy pair 108 to be sent to the remote disk system (step 600). The local disk system then takes the current value of the sequence counter 207 as the sequence number 115 that will be associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 207 is incremented (step 601). Next, the selected write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system 105 (step 602, see also Fig. 1).

In accordance with the invention, the L-write data send process 211, then proceeds to process the next write data 113 without waiting for any indication from the remote disk system 105 as to completion. In this manner, high data transfer rates are realized. However, there is the possibility of packets arriving out of sequence. Hence, the local disk system 104 checks whether there are any write data 113 that belong to the

remote disk system 105 in the remote copy pair 108 which have not yet been sent (step 603). If so, processing continues at step 600. If not, then the process waits for a while (step 604) and then continues at step 603.

The waiting in step 604 is to accommodate the situation where there is no write data to be sent. If there is no write data to send in step 603, the L-write data send process 211 has no processing to do. Consequently, step 604 is used to pause the process for a while before checking again to see if there is write data to send.

Referring now to Figs. 2 and 7, the processing in the L-write data send completion process 212 will be described. This process is executed when the local disk system 104 receives a notification of the completion of the transfer of the write data 113 to the remote disk system 105. The local disk system receives the notification of the transfer of the write data 113 from the remote disk system 105 (step 700). The local disk system 104 then makes the value of the corresponding time stamp 114 and the sequence number 115 NULL. Since the write data 113 has already been sent to the remote disk system 105, these fields are no longer used by the local disk system 104.

After the transfer of write data 113 to the remote disk system 104 is completed, the write data 113 is then written to the local disk unit 104 by the L-write data destage process 213. This process results in the actual writing of the write data 113 to the local disk unit 106 and is performed according to known conventional methods.

Referring now to Figs. 2 and 8, the R-write data receive process 214 will be discussed. This process is executed in the remote disk system 105 when it receives write data 113 from the local disk system 104. In accordance with the embodiment of the invention shown in Fig. 2, there is an R-write data receive process 214 for each remote copy pair 108. The remote disk system 105 stores the received write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 into cache memory 203 (step 800). The remote disk system 105 sends a completion indication of the transfer of the write data 113 along with its sequence number 115 back to the local disk system (step 801).

The cache memory 203 contains, for each remote copy pair 108, a data structure comprising a received write data 113, a remote disk address 208, positioning information 209, a sequence number 115, and a time stamp 114. Thus, there is one such data structure for each block of write data 113 that has been received from the local HOST 102, thereby resulting in plural such data structures.

Referring now to Figs. 2 and 9, the slave limit time schedule process 116 will be described. The slave process 116 executes for a period of time and terminates. This process is activated periodically. Recall that in the embodiment of the invention shown in Fig. 2, each remote copy pair 108 has an associated sequence number 115. In fact, for each block of write data 113 that is waiting in a remote copy pair 108, there is an associated sequence number 115. Thus, each remote copy pair 108 may have a list of sequence numbers 115. Recall further that a remote disk system 105 may have more than one remote copy pair 108 associated with it. Consequently, each remote disk system 105 may have more than one list of sequence numbers.

For each remote copy pair 108, the slave limit time schedule process 116 inspects the list of sequence numbers in that remote copy pair (step 901). It finds the longest run of numbers that are sequential and returns the maximum sequence number from that run. For example, suppose a remote copy pair contained the following list of sequence numbers:

..., 11, 12, 13,14,15,16,17,18,19,20,21, 22, 23, 24, 25, 26,28,29

In the above example, the sequence numbers which are 26 or less than 26 are sequential because the sequence number 27 has not been yet received. And so the process will select '26' which is the highest-valued sequence number in the run. Next, the process searches the time stamp 114 associated with the highest-valued (maximum) sequence number. This is repeated for each remote copy pair, resulting in a list of time stamps.

From this list of time stamps, the earliest time stamp is selected (step 901). Each slave limit time schedule process 116 produces one such earliest time stamp in this manner. The earliest time stamp from each slave process is then delivered to the master time limit schedule process 117 (step 902, see also Fig. 1). Each slave process then waits (step 903) to receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904, see also Fig. 1).

Referring now to Figs. 2 and 10, the master limit time schedule process 117 will be described. This process is activated when the remote disk system receives time stamps 114 from each of the slave processes 116. The master process 117 selects the earliest time stamp from among the received time stamps (step 1000) and sends the selected time stamp as a limit time 118 to each slave process 116 (step 1001).

Referring now to Figs. 2 and 11, the R-write data destage process 215 will be described. This process is executed when the remote disk system 105 destages write data 113 associated with a remote copy pair 108 onto the remote disk unit 107 associated with that remote copy pair 108. The remote disk system 104 selects a candidate write data 113 whose associated time stamp 114 is the earliest (step 1100). Then it compares the selected time stamp with the limit time 118 (step 1101) defined by the master limit time schedule process 117. If the selected time stamp 114 is later than the limit time 118, then the remote disk unit 105 waits for a while (step 1102) and then continues processing at step 1100. If the selected time stamp 114 is equal to or earlier than the limit time 118, then the remote disk system 105 destages (i.e., writes out to disk) the candidate write data 113 according to its associated remote disk address 208 and position information 209 (step 1103). The write data and its associated time stamp, sequence number, remote disk address information 208, and position information 209 are removed from cache memory 203 (step 1104).

The following example will be helpful. Suppose we have the following:

write data A, time stamp 10:00
write data B, time stamp 10:02
write data C, time stamp 10:04
:
:
limit time: 10:01

In this example, the R-write data destage process selects write data A, which has the earliest time stamp (10:00). Next, the R-write data destage process destages write data A to a remote disk unit 107 because its time stamp (10:00) is earlier than the limit time (10:01). After the destaging of the write data A, write data B has the earliest time stamp. However, write data B cannot be destaged because the limit time (10:01) is earlier than the write data B's time stamp(10:02). The destaging of the write data B will become possible after the limit time 118 is updated to a time later than 10:02 by the slave limit time schedule process 116 and the master limit time schedule process 117 (shown in Fig. 9 and Fig. 10).

Referring now to Fig. 3, another embodiment of the invention will be described. The system diagram of Fig. 3 is essentially the same as that shown for Fig. 2, with the following differences. In the embodiment of the invention shown in Fig. 3, there is a sequence number 115 for each remote disk system 105 that is associated with the local disk system 104. Recall in Fig. 2, there is a sequence number for each remote copy

pair 108. However, for the embodiment shown in Fig. 3, the local disk system 104 has one sequence number 115 for each remote disk 104 system which shares at least one remote copy pair 108 with that local disk system. Thus, as can be seen in Fig. 3, there is a data pair comprising a sequence counter 300 and a remote disk system address 301, for each remote disk system. Thus, if a local disk system is associated with two remote disk systems, there will be two such data pairs contained in the local disk system.

In the embodiment shown in Fig. 3, the L-write data receive process 210 is the same as described in connection with the embodiment shown in Fig. 2.

Referring now to Figs. 2 and 12, the L-write data send process 211 for the embodiment of the invention shown in Fig. 3 will be described. There is an L-write data send process 211 for each remote disk unit which shares at least one remote copy pair 108 with this local disk system 104. The local disk system 104 selects the write data 113 whose time stamp 114 is the earliest in time from all of the write data 113 belonging to the corresponding remote disk system 105 and which have not yet been sent to the remote disk system 105 (step 1200). The local disk system 104 then copies the current value of the sequence counter 300 corresponding to that remote disk system into the sequence number 115 associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 300 is incremented (step 601). Next, the selected write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system 105 (step 602). The process then continues in accordance with the description for Fig. 6.

In the embodiment of the invention shown in Fig. 3, the L-write send completion process 212 is the same as described in connection with the embodiment of the invention shown in Fig. 2.

In the embodiment shown in Fig. 3, the remote disk system 105 includes a data pair comprising a sequence counter 300 and a local disk system address 302, for every local disk system 104 which shares a remote copy pair 108 with the remote disk system 105. There is one such data pair for each local disk system that is associated with the remote disk system.

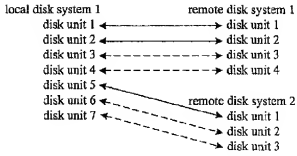
There is an R-write data receive process 214 for every local disk system which shares a remote copy pair 108 with the local disk system 105. In the embodiment of Fig. 3, the R-write data receive process 214 is the same as described in connection with the embodiment shown in Fig. 2. Similarly, the master limit time schedule process 117

and the R-write data destage process 215 are the same as described for the embodiment of the invention shown in Fig. 2.

Referring now to Fig. 13, the slave limit time schedule process 116 in accordance with the embodiment of the invention shown in Fig. 3 will be discussed. There is a slave process for each remote disk system 105. This process is essentially the same as for the embodiment of Fig. 2, with the following difference.

For each local disk system 104 that shares at least one remote copy pair 108, the slave limit time schedule process 116 inspects the list of sequence numbers received from that local disk system 104 (step 1300). It finds the maximum sequence number of the sequence numbers 114 that are sequential and returns the time stamp 115 corresponding to the maximum sequence number. This is repeated for each local disk system that shares at least one remote copy pair 108 with the remote disk system to produce a list of time stamps. From this list of time stamps 11, the earliest time stamp is selected (step 901) and delivered to the master time limit schedule process 117 (step 902). Thus, each slave limit time schedule process 116 searches one such earliest time stamp for each remote disk system 105. The process then waits (step 903) to receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904).

Referring now to Fig. 4, another embodiment of the invention will be discussed. The system diagram of Fig. 4 is essentially the same as that shown for Fig. 2, with the following differences. In the embodiment of the invention shown in Fig. 4, local disk system has a sequence number 115 for each remote disk system 105 that has a data integrity pair group 109 in common with that local disk system 104. Consider, for example, that a local disk system X has an associated data integrity pair group Z. The data integrity pair group in turn comprises five remote copy pair groups, RCP1 – RCP5. Consider further that remote copy pair groups RCP1 and RCP2 are associated with remote disk system A, remote copy pair RCP3 is associated with remote disk system B, and RCP4 and RCP5 are associated with remote disk system C. In this particular example, the local disk system X will have three sequence numbers associated with data integrity pair group Z.




The differences among the embodiments shown in Figs. 2 – 4 are illustrated in the exemplary configuration shown above, where a local disk system 104 has seven local disk units and two remote disk systems 105. The double arrow indicates a remote copy pair; e.g. local disk unit 5 and remote disk unit 1 define a remote copy pair 108. In addition, the solid-lined remote copy pairs together define one data integrity group, while the broken-lined remote copy pairs together define a second data integrity group.

Thus, in accordance with the embodiment of the invention shown in Fig. 2, the local disk system 104 would have an associated sequence number 114 for each of its disk units (a total of seven sequence numbers). In accordance with the embodiment shown in Fig. 4, the local disk system 104 has two associated sequence numbers, one for each remote disk system 105 which has a remote copy pair in common with the local disk system 105. In accordance with the embodiment shown in Fig. 4, the local disk system 104 has four associated sequence numbers 114, one for each remote disk system 105 that shares a data integrity pair group 109 with the local disk system 104. The solid-lined data integrity pair group is shared between the two remote disk systems 104; likewise, the broken-lined data integrity pair group is shared between the two remote disk systems 104.

Now continuing with Fig. 4, as can be seen, there is a data integrity pair group information data structure 400 in the local disk system 104 for each remote disk system 105 that is associated with the local disk system 104 via a data integrity pair group 109. The data structure 400 includes a data integrity group id 401 and one or more data pairs comprising a remote disk system address 403 and a sequence counter 402. There is one such data pair for each remote disk system 105 in the data integrity pair group 109. The remote disk system 105 includes a similar data structure, for each local disk system 104 that is associated with the remote disk system 105 via a data integrity pair group 108.

In the embodiment of the invention shown in Fig. 4, the L-write send completion process 210 and the L-write send completion process 212 operate in accordance with embodiment of the invention shown in Fig. 2.

Referring now to Fig. 14, the L-write data send process 211 for the embodiment of the invention shown in Fig. 4 will be described. There is an L-write data send process 211 for each remote disk unit which shares at least one data integrity pair group 109 with this local disk system 104. The local disk system 104 selects the write data 113 whose time stamp 114 is the earliest in time from all of the write data 113 belonging to the corresponding data integrity pair group 109 and which have not yet been sent to the remote disk system 105 (step 1400). The local disk system then copies the current value of the sequence counter 403 corresponding to the target remote disk system into the sequence number 115 associated with the selected write data 113 (i.e., the write data whose associated time stamp is the earliest in time). The sequence counter 403 is incremented (step 601). Next, the selected write data 113 and its associated time stamp 114, sequence number 115, remote disk address information 208, and position information 209 are sent to the remote disk system (step 602). The process then continues in accordance with the discussion for Fig. 6.

In the embodiment of the invention shown in Fig. 4, an R-write data receive process 214 is associated with each local disk unit 104 that shares a data integrity pair group 109 associated with this remote disk system 105. Processing proceeds in accordance with the flow chart shown in Fig. 8.

The master limit time schedule process 117 and the R-write data destage process 215 operate in the same manner as described for the embodiment of the invention shown in Fig. 2.

Referring now to Fig. 15, the slave limit time schedule process 116 in accordance with the embodiment of the invention shown in Fig. 4 will be discussed. There is a slave process for each remote disk system 105. This process is essentially the same as for the embodiment of Fig. 2, with the following difference.

For each local disk system 104 that is associated with the remote disk system via a data integrity pair group, the slave limit time schedule process 116 inspects the list of sequence numbers received from that local disk system (step 1500). It finds the maximum sequence number of the sequence numbers 114 that are sequential and returns the time stamp 115 corresponding to the maximum sequence number . This is repeated for each local disk system 104 associated with the remote disk system via a data integrity

pair group 109 to produce a list of time stamps. From this list of time stamps, the earliest time stamp is selected (step 901) and delivered to the master time limit schedule process 117 (step 902). Thus, each slave limit time schedule process 116 searches one such earliest time stamp for each remote disk system 105. The process then waits (step 903) to receive a limit time value 118 from the master limit time schedule process 117 and stores the value in cache 202 (step 904).

Turn now to Fig. 16 for a discussion of an embodiment of the invention in the context of magnetic tape (MT) storage systems deployed at the remote site. A local computer system 1600 includes at least a local HOST system 1602 and at least a local storage system 1604 such as a local disk system. The local computer system may include a switch system 1606. There is a remote computer system 1601 that includes at least a remote MT system 1605. The remote HOST system 1603 is not always present in the remote computer system 1601.

The local HOST system 1601, the local disk system 1604, and the switch system 1606 are connected to the remote disk system 1605 via a communication path 1613. The communication path 1613 is a network which is characterized in that transmitted data packets do not necessarily arrive in the order in which they were sent. For example, an IP-based network exhibits this behavior. In this embodiment of the invention, a unit of read/write data is called a block. Each block has a block id which is used to identify itself.

A local storage system 1604 has at least one local storage unit 1607 and a storage control unit 1609 with a cache memory 1611. The remote MT system 1605 includes at least one remote MT unit 1608 and an MT control unit 1610 with a cache memory 1611. The local HOST system 1602 has a main memory 1612. The switch system 1606 also includes a cache memory 1611.

In accordance with the embodiment of the invention shown in Fig. 16, a read request and a write request to a remote MT unit 1608 each includes a sequence number 1614 that accompanies the data to be read 1624 or written 1625.

Fig. 16 also shows the processes of the local HOST system 1602, the local disk system 1604, the remote MT system 1605, and the switch system 1606. A read request issue process 1615 and a write request issue process 1616 are provided in the local HOST system 1602, the local disk system 1604, and the switch system 1606. The remote MT system 1605 includes a read request receive process 1617 and a write request receive process 1618.

Processing of the read request issue process 1615 is outlined in Fig. 17. This flow is common among the local HOST system 1602, the local storage system 1604 and the switch system 1606. A sequence counter 1623 that is copied to cache memory 1611 of the local storage control unit 1609 (or into main memory 1612) provides the value for sequence number 1614 when a read/write request is issued (step 1700). The process then issues a read request to the remote MT system 1605 (step 1701). The process then increments the sequence counter 1623 (step 1702).

Next (step 1703), the read request issue process checks the number of requests that have been issued. If the number is less than a value m, processing continues at step 1700 in order to issue a next request. The value for m is typically > 2, and preferably is determined empirically. The goal is to obtain better performance by sending the next request before the completion of the present request.

If the number is not less than m, then the process waits for a data transfer notification from the remote MT system 1605 (step 1704). When the read request issue process 1615 receives a data transfer notification from the remote MT system 1605, it receives the read data 1624 and the sequence number 1614 and stores it into cache memory 1611 (or into main memory 1612) (step 1705) according to the sequence counter 1623. Next, the read request issue process 1615 checks if it has received all the data transfer notifications from the remote MT system 1605 (step 1706). If not, the process continues at step 1703 to wait for the data transfer notification.

Processing of the write request issue process 1616 is outlined in the flowchart shown in Fig. 18. This flow is common among the local HOST system 1602, the local storage system 1604, and the switch system 1606. The write request issue process copies the contents of the sequence counter 1623 to the sequence number 1614 into a cache memory 1611 (or into main memory 1612) (step 1800), and issues a write request with the sequence number 1614 and write data 1625 to the remote MT system 1605 (step 1801). Then it increments the sequence counter 1623 (step 1802). In step 1803, the write request issue process 1616 checks the number of requests issued. If the number is less than the value m, it jumps to step 1800 to issue a next request. If not so, it waits for the notification of the data transfer from the remote MT system (step 1804). When the write request issue process 1616 receives the notification from the remote MT system 1605 (step 1805), it then checks if it has received all the data transfer notification from a remote MT system 1605 (step 1806). If not so, it jumps to step 1803 to wait for the data transfer notification.

The flow of the read request receive process 1617 in the remote MT system 1605 is shown in Fig. 19. The remote MT system 1605 has the read data 1624 in the cache memory 1611 of the MT control unit 1610. A group of read data 1624 is read from one remote MT unit 1608 and stored into an MT queue 1626. Each queue entry is a data pair comprising the read data 1624 and its corresponding block id information 1627. Read data 1624 and its block id information 1627 are inserted into the queue according to the block id information 1627. The MT queue 1626, therefore, is ordered by block id information 1627. This ordering represents the order in which the data blocks should be sent back to the requesting local HOST 1601. Thus, a series of read requests, if received in proper sequence, would be satisfied simply by sending each data block in the order that they occur in the MT queue 1626.

The queue 1626 is a doubly-linked data structure. Each queue entry therefore further includes a forward pointer 1630 which points to a queue entry including next block id information 1627 and a backward pointer 1631 which points to a queue entry including previous block id information 1627. A head pointer 1628 points to the head of the MT queue 1626 and a tail pointer 1629 points to the tail of the MT queue 1626. A next sequence number counter 1632 contains the sequence number 1614 corresponding to the read data 1624 that should be accessed by a next read request. Hence, the current value of the counter 1632 corresponds to the data block at the head of the MT queue.

In operation, the read request receive process 1617 receives, from a local HOST 1601, a read request with a sequence number 1614 (step 1900). The read request receive process compares the received sequence number 1614 to the next sequence number counter 1632 (step 1901). If it is equal, the process sends the received sequence number 1614 and the read data 1624 at the head of the MT queue 1626 (pointed to by head pointer 1628), along with the received sequence number 1614 to a read request issue process 1615 (step 1902). The sequence number counter 1632 is then updated, referring to the block id information 1627 of the next read data 1624 in the MT queue (step 1903). The process then continues at step 1907.

If the received sequence number is not equal to the next sequence number counter 1632, this means that an out-of-sequence read request has been received. In step 1904, the read request receive process 1617 calculates the block id information 1627 of the read data to be requested. Specifically, the read request receive process obtains the difference D between the next sequence number counter 1632 and the received sequence

number 1614 and adds the difference D to the block ID information B of the read data 1624 at the top of the MT queue 1626. That is, B+D is the block id information 1627 to be found. In step 1905, the read request receive process searches the queue entry whose block id information 1627 is B+D. The read data 1624 at that entry in the queue 1626, along with the received sequence number 1614, is sent to the read request issue process 1615 (step 1906). Processing continues to step 1907.

In step 1907, the read request receive process 1617 deletes the queue entry corresponding to the sent read data 1624. Thus, by using a sequence number 1614, the remote MT system 1605 can recognize the accessed read data 1624 without mistake even if the read requests becomes out of sequence due to the nature of the communication path 1613. Referring back to Fig. 17, the received data is assembled according to the received sequence number 1614. Thus, if the read requests at the remote system get out of sequence, the sequence number ensures that they are satisfied in correct order. Likewise, if the received data at the local system get out of order, it is assembled correctly because of the sequence number 1614.

The flow of the write request receive process 1618 in the remote MT system 1605 is shown in Fig. 20. The structure of the MT queue 1626 for data to be written is same as that of the read data 1624. In the case of write data 1625, a next sequence number counter 1632 in the MT unit 1605 is greater than the sequence number 1614 of the write data pointed to by tail pointer 1629 by one.

A write request receive process 1618 at the remote computer receives a write request with sequence number 1614 and write data 1625 , and stores the write data into cache memory 1611 (step 2000). Next, the write request receive process 1618 checks whether the sequence number 1614 is equal to or more than a next sequence number counter 1632 (step 2001). If equal, it inserts the received write data 1625 at the tail of the corresponding write MT queue 1626, thus updating the information in the MT queue (step 2002). It then calculates the block id of received write data 1625 and stores it into block id information 1627 (step 2003). Then it adds D2 (the difference between the sequence number 1614 and the next sequence number counter 1632) + 1 to the next sequence number counter 1632 (step 2004) and jumps to step 2009.

If the received sequence number is less than the next sequence number counter 1632, then it computes the difference between the next sequence number counter 1632 and the received sequence number 1614 (step 2005). It finds the position of the MT queue 1626 to insert the write data 1624, thus updating the information in the MT queue

(step 2007). Then it calculates a block id of the received write data 1625 and stores it into block id information 1627 (step 2008). Next, the write request receive process notifies the completion of the write request and sends back the received sequence number 1614 to the write request issue process 1616 (step 2009). Thus, by using a sequence number 1614, the remote MT system 1605 can also recognize the block id of the write data 1625 without mistake, even if the write request gets out of sequence due to the communication path 1613.

Fig. 21 shows another embodiment of the present invention for MT storage systems. This embodiment of the invention is essentially the same as the embodiment shown in Fig. 15. The difference is the use of a next block id information 1700, 1701 in place of the sequence number 1614 used in Fig. 16 to detect out of sequence occurrences and to correct for such occurrences.

WHAT IS CLAIMED IS:

1. A storage system comprising:

a first computer system having a first storage component; and

a second computer system having a second storage component,

the first and second storage components configured to exchange data over a data network,

the first computer system having a memory that is configured with program code to write a block of data to the first storage component and to transmit a data packet to the second computer system, the data packet including the block of data, a time stamp, and a sequence number,

the second computer system having a memory that is configured with program code to receive data packets from the first computer system, to select a candidate data packet based on time stamps and sequence numbers contained in the data packets, and to write the candidate data packet on the second storage system,

wherein blocks of data written on the first storage component are written on the second storage component in the same order as on the first storage component.

2. The system of claim 1 wherein the second memory is further configured with program code to obtain a limit time stamp from among the time stamps based on their corresponding sequence numbers and to select the candidate data packet from among the data packets by comparing their corresponding time stamps against the limit time stamp.

3. The system of claim 1 wherein the data network is a connectionless network.

4. The system of claim 1 wherein the data network is characterized as being unable to guarantee that data packets will be received in the same order as they were sent.

5. The system of claim 4 wherein the data network is a wide area network.

6. The system of claim 1 wherein the first storage component comprises plural first data storage units, and the second storage component comprises

plural second data storage units, each of the first data storage units corresponding to one of the second data storage units, wherein data stored on one of the first data storage units is also stored on the corresponding second data storage unit.

7. The system of claim 1 wherein the first storage component comprises plural first disk systems and the second storage component comprises plural second disk systems, each first disk system being associated with one or more of the second disk systems, wherein data stored in one of the first disk systems is also stored on the associated one or more of the second disk systems.

8. The system of claim 7 wherein each of the first disk systems comprises plural first disk units and each of the second disk systems comprises plural second disk units, each of the first disk units being associated with one of the second disk units.

9. The system of claim 8 wherein each first disk unit is associated with one of the second disk units independently of the first disk system to which the first disk unit belongs.

10. A method of backing up data contained in a local system to a remote system, comprising:

writing a block of data to a local data store;

sending a data packet to the remote system, the data packet including the block of data, a time stamp, and a sequence number;

receiving data packets from the local system; and

selecting a data packet whose block of data is to be written on a remote data store, based on the sequence numbers and the time stamps of the data packets.

11. The method of claim 10 further including incrementing the sequence number for a next data packet.

12. The method of claim 10 wherein selecting a data packet includes obtaining a limit time stamp from among the time stamps based on their associated sequence numbers and selecting the data packet from among the data packets by comparing their associated time stamps against the limit time stamp.

13. The method of claim 10 wherein the local data store comprises plural local disk units and the remote data store comprises plural remote disk units, each local disk unit being paired with one of the remote disk units to define a remote copy pair.

14. The method of claim 13 further including writing plural blocks of data to the local disk units and sending plural data packets to the remote disk units so that each remote disk unit has a list of sequence numbers from its associated plural data packets, the method further including, for each list of sequence numbers, obtaining a longest run of sequence numbers, obtaining the highest-valued sequence number from the longest run, and obtaining the time stamp corresponding to the highest-valued sequence number, thereby producing a list of time stamps, the method further including selecting a data packet based on the earliest time stamp in the list of time stamps.

15. The method of claim 10 wherein the local data store comprises plural local disk systems and the remote data store comprises plural remote disk systems, each local disk system being associated with one or more of the remote disk systems, wherein data stored in one of the local disk systems is also stored on the associated one or more of the remote disk systems.

16. The method of claim 15 wherein each of the local disk systems comprises plural local disk units and each of the remote disk systems comprises plural remote disk units, each of the local disk units being associated with one of the remote disk units.

18. The method of claim 16 wherein each local disk unit is associated with one of the remote disk units independently of the local disk system to which the local disk unit belongs.

19. The method of claim 10 wherein writing a block of data to a local data store and sending a data packet to the remote system are performed asynchronously.

20. The method of claim 10 wherein the data packets are sent over a connectionless data network.

21. The method of claim 10 wherein the data packets are sent over a data network that is characterized as being unable to guarantee that data packets will arrive at a destination in the same order as they were sent.

22. The method of claim 21 wherein the data network is a wide area network.

23. In a local storage system comprising plural local data stores, a method for backing up data in the local storage system to a remote storage system comprising plural remote data stores, the method comprising:

each local data store, receiving a data block to be written thereto;

each local data store transmitting a data packet comprising the data block, a time stamp, and a sequence number to one of the remote data stores;

at the remote data stores, receiving plural data packets from the local data stores, wherein each remote data store has its associated plural data packets and a list of sequence numbers and a list of time stamps from the associated data packets;

at each remote data store, identifying a longest run of sequence numbers and obtaining the data packet of the highest-valued sequence number of the longest run;

at each remote data store, obtaining the earliest time stamp from the obtained data packet;

selecting the earliest of the obtained time stamps as a limit time;

at each remote data store, selecting a candidate data packet having the earliest time stamp; and

selecting the data packet from among the candidate data packets whose time stamp is earlier than the limit time.

24. The method of claim 23 wherein the local storage system comprises one or more local disk systems, each local disk system comprises one or more local disk drives, the remote storage system comprises one or more remote disk systems, and each remote disk system comprises one or more remote disk drives, each local disk drive being associated with one of the remote disk drives to define a remote copy pair.

25. The method of claim 24 wherein each local data store is one of the local disk drives and each remote data store is one of the remote disk drives, wherein there is a sequence number associated with each remote copy pair.

26. The method of claim 25 wherein the received plural data packets are grouped according to remote copy pair.

27. The method of claim 24 wherein a sequence number is associated with each pair of local and remote disk systems which have a common remote copy pair.

28. The method of claim 27 wherein each of the plural data packets is grouped based on the local disk system from which it was sent.

29. The method of claim 24 wherein each remote copy pair is associated with one of a plurality of data integrity pair groups, wherein a sequence number is associated with each pair of local and remote disk systems which have a common data integrity pair group.

30. A data access method comprising:

providing a data transfer request, the data transfer request including a sequence number;

transmitting the data transfer request from a local system to a remote system;

at the remote system, providing a queue of entries containing blocks of data, the data transfer request being directed to a target entry in the queue;

at the remote system, comparing the sequence number in the data transfer request against the current value of a sequence number counter;

if the sequence number is not equal to the sequence number counter, then gaining access to the target entry in the queue by traversing the queue by a number of entries based on a difference between the sequence number and the sequence number counter;

if the sequence number is equal to the sequence number counter, then accessing one end of the queue to gain access to the target entry; and

executing the data transfer request on the target entry.

31. The method of claim 30 wherein the remote system includes a magnetic tape storage system and the data transfer requests are read and write requests to the magnetic tape storage system.

32. The method of claim 30 wherein the data transfer request is a read request and the blocks of data in the queue are used to satisfy the read request.

33. The method of claim 30 wherein the data transfer request is a write request which includes write data to be inserted into the queue as a new entry, the new entry being inserted before or after the target entry.

[図１]

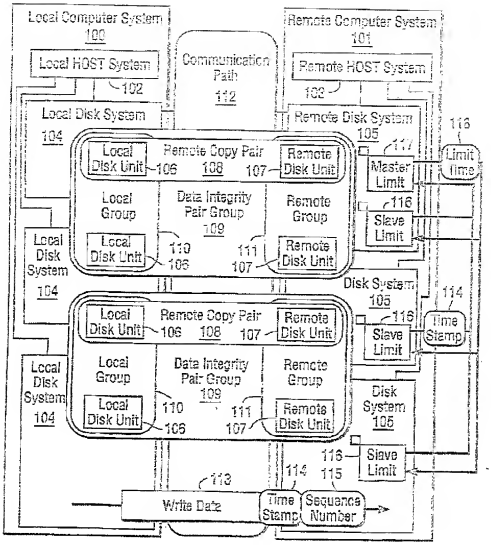

Local Computer System
100
Local HOST System
102
Communication Path
112
Remote Computer System
101
Remote HOST System
103
Local Disk System
104
Remote Disk System
105
Local Disk Unit
106
Remote Copy Pair
108
Remote Disk Unit
107
Master Limit
Limit Time
Local Group
Data Integrity Pair Group
109
Remote Group
Slave Limit
116
Time Stamp
114
Write Data
Sequence Number
113
115

FIG. 1

[図２]

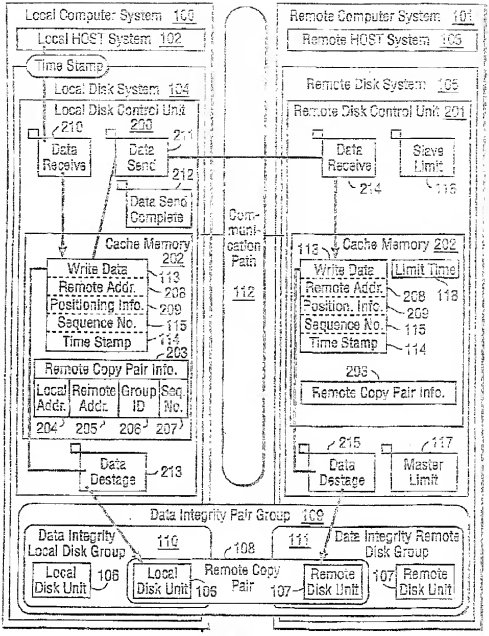

Local Computer System 100
Local HOST System 102
Time Stamp
Local Disk System 104
Local Disk Control Unit 200
210
Data Receive
Data Send
212
Data Send Complete
Cache Memory 202
Write Data 113
Remote Addr. 208
Positioning Info. 209
Sequence No. 115
Time Stamp 114
203
Remote Copy Pair Info.
Local Addr. 204
Remote Addr. 205
Group ID 206
Seq. No. 207
Data Destage 213
Communication Path 112
Remote Computer System 101
Remote HOST System 103
Remote Disk System 105
Remote Disk Control Unit 201
Data Receive 214
Sleep Limit 116
Cache Memory 202
Write Data 113
Limit Time 118
Remote Addr. 208
Position. Info. 209
Sequence No. 115
Time Stamp 114
Remote Copy Pair Info. 203
215
Data Destage
217
Master Limit
Data Integrity Pair Group 109
Data Integrity Local Disk Group 110
Local Disk Unit 106
Local Disk Unit 106
Remote Copy Pair 107
108
111
Data Integrity Remote Disk Group
Remote Disk Unit 107
Remote Disk Unit

FIG. 2

[図３]

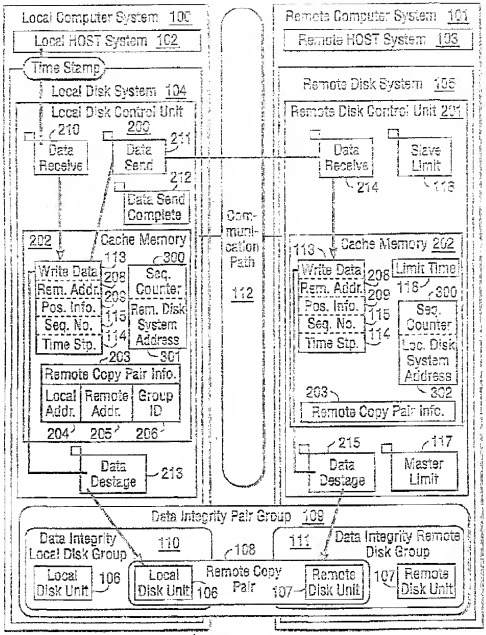

Local Computer System 100
Local HOST System 102
Time Stamp
Local Disk System 104
Local Disk Control Unit 200
210
Data Receive
Data Send
211
212
Data Send Complete
202
Cache Memory 300
Write Data 208
Rem. Addr. 209
Pos. Info. 115
Seq. No. 114
Time Stp. 203
Sec. Counter
Rem. Disk System Address 301
Remote Copy Pair Info.
Local Addr. 204
Remote Addr. 205
Group ID 206
Data Destage 213
Communication Path 112
Remote Computer System 101
Remote HOST System 103
Remote Disk System 105
Remote Disk Control Unit 201
Data Receive 214
Slave Limit 116
113
Cache Memory 202
Write Data 208
Rem. Addr. 209
Pos. Info. 115
Seq. No. 114
Time Stp. 203
Limit Time 118
300
Seq. Counter
Loc. Disk System Address 302
Remote Copy Pair Info.
215
Data Destage
117
Master Limit
Data Integrity Pair Group 109
Data Integrity Local Disk Group 110
Local Disk Unit 106
Local Disk Unit 106
108
Remote Copy Pair
111
Data Integrity Remote Disk Group
Remote Disk Unit 107
Remote Disk Unit 107

FIG. 3

[図４]

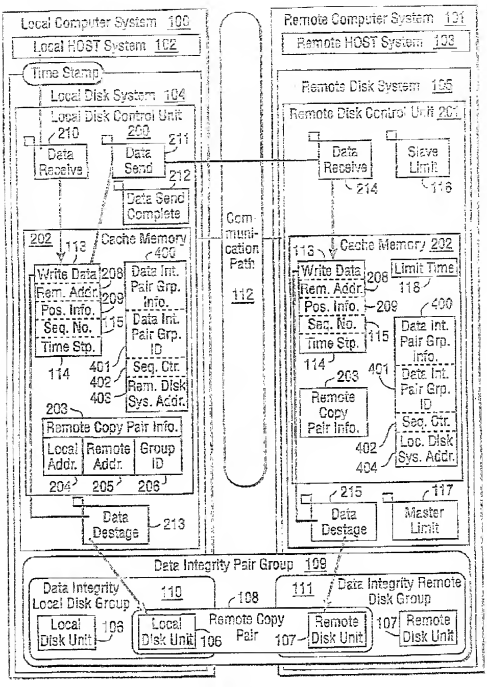

Local Computer System 100
Local HOST System 102
Time Stamp
Local Disk System 104
Local Disk Control Unit 200
210
Data Receive
Data Send
211
212
Data Send Complete
202
113
Cache Memory 400
Write Data 208
Rem. Addr. 209
Pos. Info.
Seq. No. 115
Time Stp.
114
401
402
403
Data Int. Pair Grp. Info.
Data Int. Pair Grp. ID
Seq. Ctr.
Rem. Disk Sys. Addr.
203
Remote Copy Pair Info.
Local Addr. 204
Remote Addr. 205
Group ID 206
Data Destage 213
Communication Path 112
Remote Computer System 101
Remote HOST System 103
Remote Disk System 105
Remote Disk Control Unit 201
Data Receive
Slave Limit
214
116
113
Cache Memory 202
Write Data
Rem. Addr.
Pos. Info.
Seq. No.
Time Stp.
206
209
115
114
203
401
402
404
Limit Time 118
400
Data Int. Pair Grp. Info.
Data Int. Pair Grp. ID
Seq. Ctr.
Remote Copy Pair Info.
Loc. Disk Sys. Addr.
215
117
Data Destage
Master Limit
Data Integrity Pair Group 109
Data Integrity Local Disk Group 110
Local Disk Unit 106
108
Local Disk Unit 106
Remote Copy Pair 107
111
Data Integrity Remote Disk Group
Remote Disk Unit
107
Remote Disk Unit

FIG. 4

[図５]

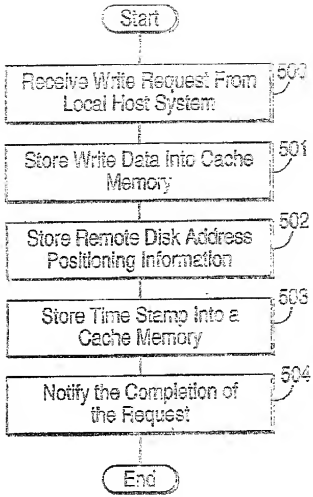

Start
Receive Write Request From Local Host System
500
Store Write Data Into Cache Memory
501
Store Remote Disk Address Positioning Information
502
Store Time Stamp Into a Cache Memory
503
Notify the Completion of the Request
504
End

FIG. 5

[図６]

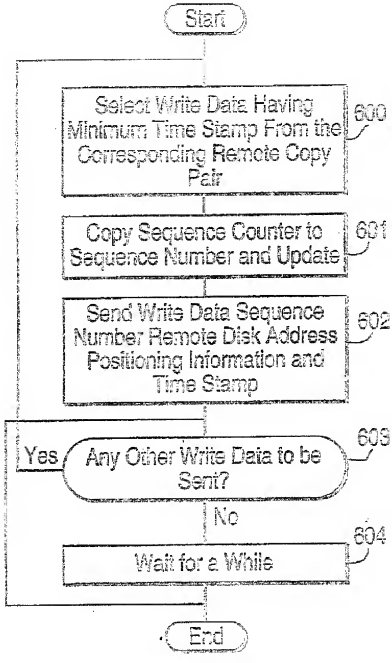

Start
Select Write Data Having Minimum Time Stamp From the Corresponding Remote Copy Pair
600
Copy Sequence Counter to Sequence Number and Update
601
Send Write Data Sequence Number Remote Disk Address Positioning Information and Time Stamp
602
Yes
Any Other Write Data to be Sent?
603
No
Wait for a While
604
End

FIG. 6

【図７】

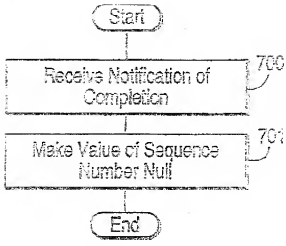



FIG. 7

【図８】

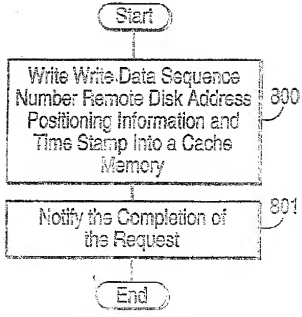



FIG. 8

[図９]

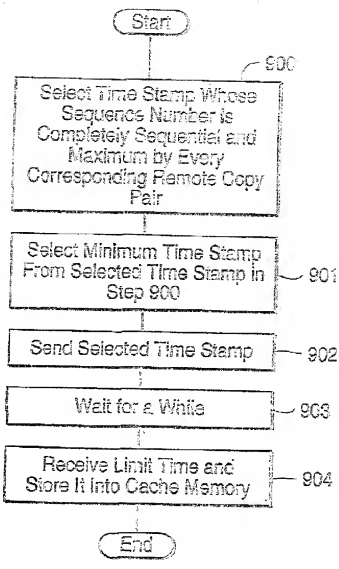

Start
900
Select Time Stamp Whose Sequence Number is Completely Sequential and Maximum by Every Corresponding Remote Copy Pair
Select Minimum Time Stamp From Selected Time Stamp in Step 900
901
Send Selected Time Stamp
902
Wait for a While
903
Receive Limit Time and Store It Into Cache Memory
904
End

FIG. 9

【図１０】

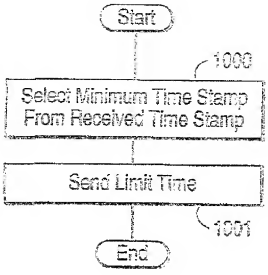

Start
1000
Select Minimum Time Stamp From Received Time Stamp
Send Limit Time
1001
End

FIG. 10

[図１１]

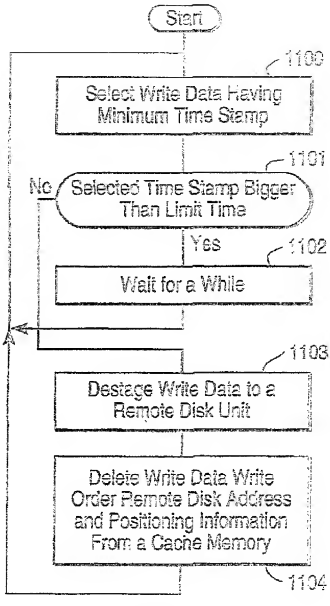

Start
1100
Select Write Data Having Minimum Time Stamp
1101
No
Selected Time Stamp Bigger Than Limit Time
Yes
1102
Wait for a While
1103
Destage Write Data to a Remote Disk Unit
Delete Write Data Write Order Remote Disk Address and Positioning Information From a Cache Memory
1104

FIG. 11

[図１２]

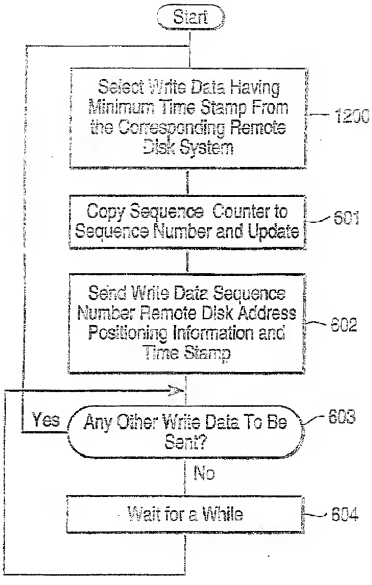

Start
Select Write Data Having Minimum Time Stamp From the Corresponding Remote Disk System
1200
Copy Sequence Counter to Sequence Number and Update
601
Send Write Data Sequence Number Remote Disk Address Positioning Information and Time Stamp
602
Yes
Any Other Write Data To Be Sent?
603
No
Wait for a While
604

FIG. 12

[図１３]

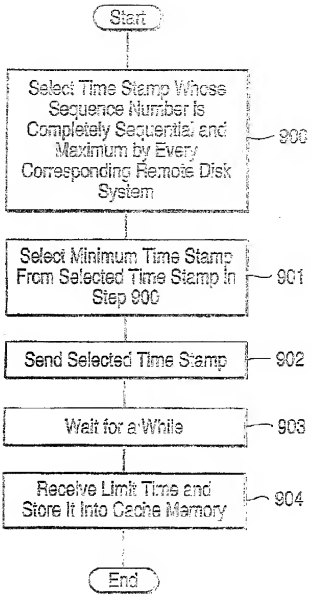

Start
Select Time Stamp Whose Sequence Number is Completely Sequential and Maximum by Every Corresponding Remote Disk System
900
Select Minimum Time Stamp From Selected Time Stamp In Step 900
901
Send Selected Time Stamp
902
Wait for a While
903
Receive Limit Time and Store It Into Cache Memory
904
End

FIG. 13

[図14]

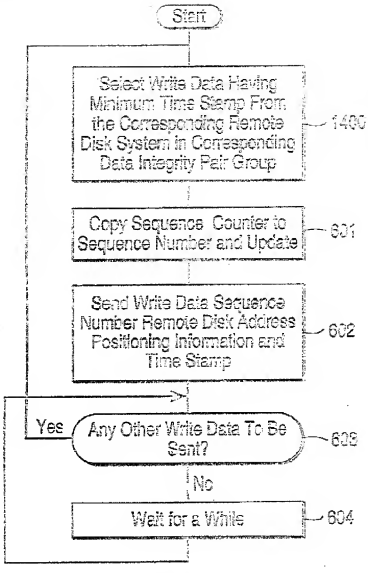

Start
Select Write Data Having Minimum Time Stamp From the Corresponding Remote Disk System in Corresponding Data Integrity Pair Group
1400
Copy Sequence Counter to Sequence Number and Update
601
Send Write Data Sequence Number Remote Disk Address Positioning Information and Time Stamp
602
Yes
Any Other Write Data To Be Sent?
603
No
Wait for a While
604

FIG. 14

[図１５]

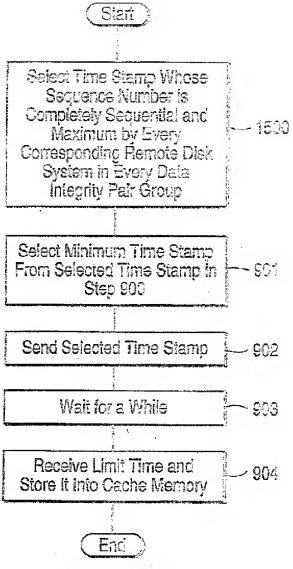

Start
Select Time Stamp Whose Sequence Number is Completely Sequential and Maximum by Every Corresponding Remote Disk System in Every Data Integrity Pair Group
1500
Select Minimum Time Stamp From Selected Time Stamp in Step 900
901
Send Selected Time Stamp
902
Wait for a While
903
Receive Limit Time and Store It Into Cache Memory
904
End

FIG. 15

[図16]

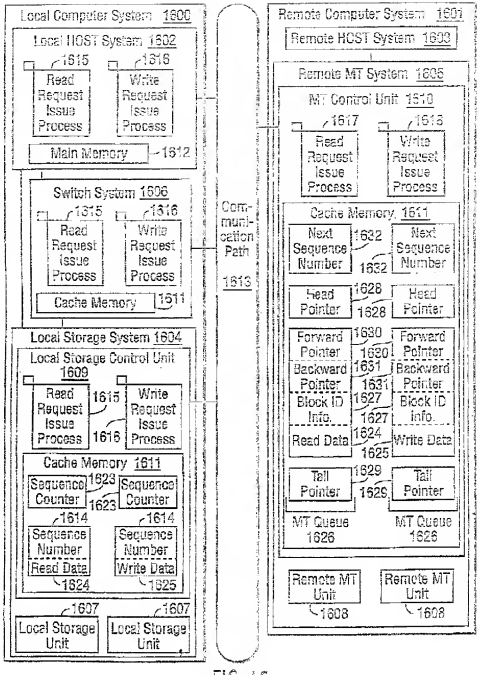

Local Computer System 1600
Local HOST System 1602
1615
Read Request Issue Process
1616
Write Request Issue Process
Main Memory 1612
Switch System 1606
1615
Read Request Issue Process
1616
Write Request Issue Process
Cache Memory 1611
Local Storage System 1604
Local Storage Control Unit 1609
Read Request Issue Process 1615
Write Request Issue Process 1616
Cache Memory 1611
Sequence Counter 1623
Sequence Counter 1623
1614
Sequence Number Read Data 1624
1614
Sequence Number Write Data 1625
1607
Local Storage Unit
1607
Local Storage Unit
Communication Path 1613
Remote Computer System 1601
Remote HOST System 1603
Remote MT System 1605
MT Control Unit 1610
1617
Read Request Issue Process
1618
Write Request Issue Process
Cache Memory 1611
Next Sequence Number 1632
1632
Next Sequence Number
Head Pointer 1628
1628
Head Pointer
Forward Pointer 1630
1630
Forward Pointer
Backward Pointer 1631
1631
Backward Pointer
Block ID Info. 1627
1627
Block ID Info.
Read Data 1624
1625
Write Data
Tail Pointer 1629
1629
Tail Pointer
MT Queue 1626
MT Queue 1626
Remote MT Unit
1608
Remote MT Unit
1608

FIG. 16

[図１７]

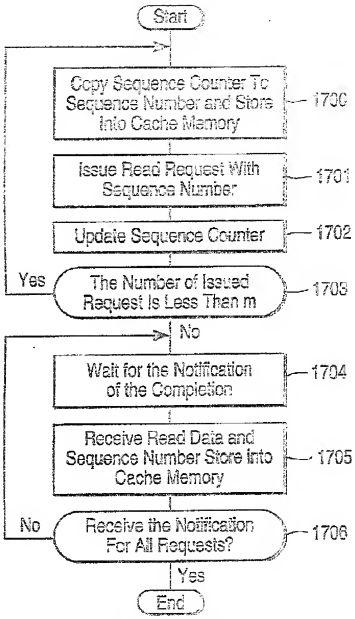

Start
Copy Sequence Counter To Sequence Number and Store Into Cache Memory
1700
Issue Read Request With Sequence Number
1701
Update Sequence Counter
1702
Yes
The Number of Issued Request Is Less Than m
1703
No
Wait for the Notification of the Completion
1704
Receive Read Data and Sequence Number Store Into Cache Memory
1705
No
Receive the Notification For All Requests?
1706
Yes
End

FIG. 17

[図18]

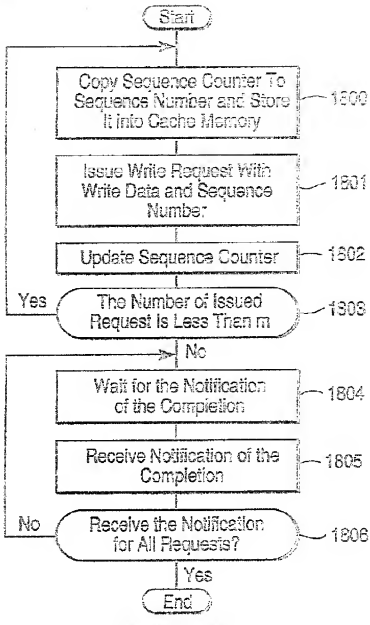

Start
Copy Sequence Counter To Sequence Number and Store It into Cache Memory
1800
Issue Write Request With Write Data and Sequence Number
1801
Update Sequence Counter
1802
Yes
The Number of Issued Request Is Less Than m
1803
No
Wait for the Notification of the Completion
1804
Receive Notification of the Completion
1805
No
Receive the Notification for All Requests?
1806
Yes
End

FIG. 18

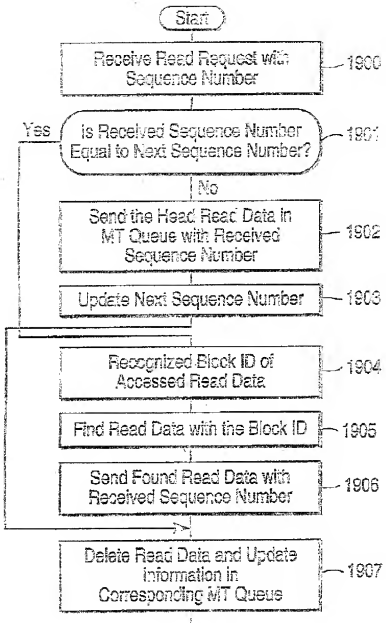

Start
Receive Read Request with Sequence Number
1900
Is Received Sequence Number Equal to Next Sequence Number?
1901
Yes
No
Send the Head Read Data in MT Queue with Received Sequence Number
1902
Update Next Sequence Number
1903
Recognized Block ID of Accessed Read Data
1904
Find Read Data with the Block ID
1905
Send Found Read Data with Received Sequence Number
1906
Delete Read Data and Update Information in Corresponding MT Queue
1907
End

FIG. 19

[도 20]

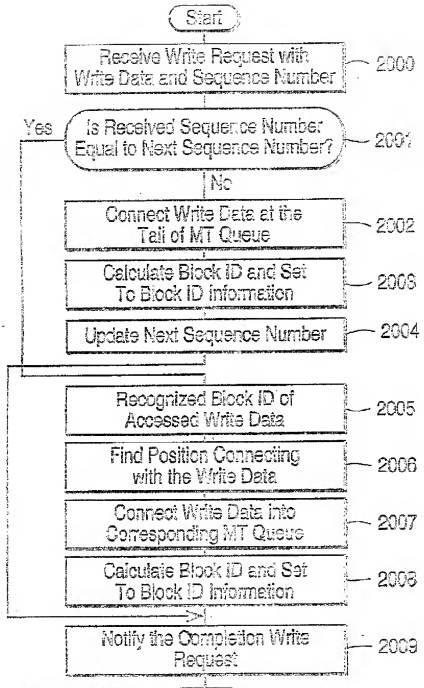

Start
Receive Write Request with Write Data and Sequence Number
2000
Yes
Is Received Sequence Number Equal to Next Sequence Number?
2001
No
Connect Write Data at the Tail of MT Queue
2002
Calculate Block ID and Set To Block ID Information
2003
Update Next Sequence Number
2004
Recognized Block ID of Accessed Write Data
2005
Find Position Connecting with the Write Data
2006
Connect Write Data into Corresponding MT Queue
2007
Calculate Block ID and Set To Block ID Information
2008
Notify the Completion Write Request
2009
End

FIG. 20

[図２１]

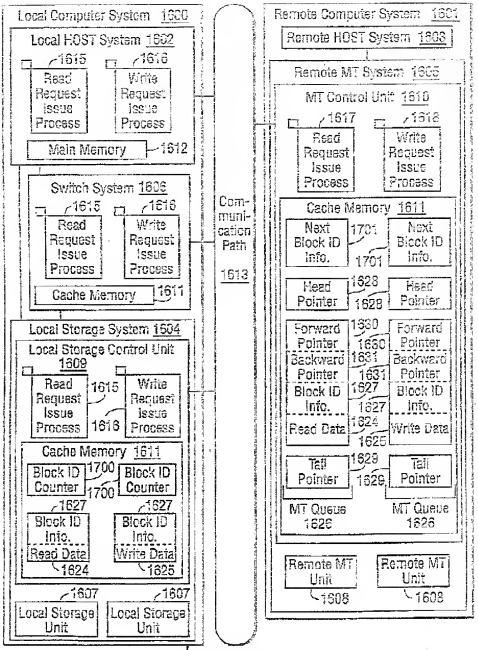

Local Computer System 1600
Local HOST System 1602
1615
Read Request Issue Process
1616
Write Request Issue Process
Main Memory 1612
Switch System 1606
1615
Read Request Issue Process
1616
Write Request Issue Process
Cache Memory 1611
Local Storage System 1604
Local Storage Control Unit 1609
Read Request Issue Process
1615
Write Request Issue Process
1616
Cache Memory 1611
Block ID Counter
1700
Block ID Counter
1700
1627
Block ID Info.
Read Data
1624
1627
Block ID Info.
Write Data
1625
1607
Local Storage Unit
1607
Local Storage Unit
Communication Path
1613
Remote Computer System 1601
Remote HOST System 1603
Remote MT System 1605
MT Control Unit 1610
1617
Read Request Issue Process
1618
Write Request Issue Process
Cache Memory 1611
Next Block ID Info.
1701
1701
Next Block ID Info.
Head Pointer
1628
1628
Head Pointer
Forward Pointer
1630
1630
Forward Pointer
Backward Pointer
1631
1631
Backward Pointer
Block ID Info.
1627
1627
Block ID Info.
Read Data
1624
1625
Write Data
Tail Pointer
1629
1629
Tail Pointer
MT Queue 1626
MT Queue 1626
Remote MT Unit
1608
Remote MT Unit
1608

FIG. 21

STORAGE SYSTEM CONNECTED TO A DATA NETWORK WITH DATA INTEGRITY

ABSTRACT OF THE DISCLOSURE

A storage system includes a local storage component coupled to a remote storage component over a communication network. The communication network is characterized by its inability to guarantee receipt of data packets in the same order the data packets are sent. A method in accordance with the invention ensures the proper ordering of handling received requests despite the nature of the communication network.